

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

#2
Phong
Dyer

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Teruhiko KAMIGATA, et al.

Application No.:

Group Art Unit:

Filed: January 24, 2002

Examiner:

For: INSTRUCTION PROCESSING METHOD FOR VERIFYING BASIC INSTRUCTION
ARRANGEMENT IN VLIW INSTRUCTION FOR VARIABLE LENGTH VLIW
PROCESSOR

J1011 U.S. PTO
10/053707
01/24/02

MA
3/22/02

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)
herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2001-128368

Filed: April 25, 2001

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: January 24, 2002

By: 

H. J. Staas

Registration No. 22,010

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

日 本 国 特 許 庁
JAPAN PATENT OFFICE

J1011 U.S. PTO
10/053707
01/24/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 4月25日

出 願 番 号

Application Number:

特願2001-128368

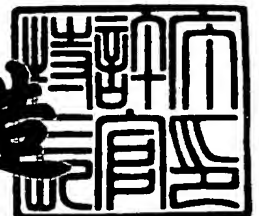
出 願 人
Applicant(s):

富士通株式会社

2001年 8月17日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3074023

【書類名】 特許願

【整理番号】 0140148

【提出日】 平成13年 4月25日

【あて先】 特許庁長官 及川 耕造 殿

【国際特許分類】 G06F 9/38

【発明の名称】 命令処理方法

【請求項の数】 10

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 上方 輝彦

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 三宅 英雄

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100070150

【住所又は居所】 東京都渋谷区恵比寿4丁目20番3号 恵比寿ガーデンプレイスタワー32階

【弁理士】

【氏名又は名称】 伊東 忠彦

【電話番号】 03-5424-2511

【手数料の表示】

【予納台帳番号】 002989

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704678

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 命令処理方法

【特許請求の範囲】

【請求項 1】 プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第 1 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 2 のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第 3 のステップと

を有する命令処理方法。

【請求項 2】 プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第 1 のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第 2 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 3 のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの物理スロットに配置する第 4 のステップと

を有する命令処理方法。

【請求項 3】 プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第 1 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 2 のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する第 3 のステップと

を有する命令処理方法。

【請求項 4】 プロセッサが実行する命令を構成する基本命令の命令カテゴ

りを特定する第 1 のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第 2 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 3 のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する配置する第 4 のステップと

を有する命令処理方法。

【請求項 5】 前記第 2 のステップは、配置可能な最も若い論理命令スロットを特定するステップを有する請求項 1 又は 3 に記載の命令処理方法。

【請求項 6】 前記第 3 のステップは、配置可能な最も若い論理命令スロットを特定するステップを有する請求項 2 又は 4 に記載の命令処理方法。

【請求項 7】 コンピュータに、

プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第 1 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 2 のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第 3 のステップと

を実行させるためのプログラム。

【請求項 8】 コンピュータに、

プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第 1 のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第 2 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 3 のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令

との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する配置する第4のステップと
を実行させるためのプログラム。

【請求項9】 コンピュータに、
プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第1のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第2のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する第3のステップと

を実行させるためのプログラム。

【請求項10】 コンピュータに、
プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第1のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第2のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第3のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する配置する第4のステップと

を実行させるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

近年の計算機システムの処理能力の向上は、中央処理装置（CPU: Central Processing Unit）であるプロセッサの処理能力の向上により為されてきた。このプロセッサの処理能力を向上させる手法としては、命令レベル並列度（ILP: Instruction Level Parallelism）を向上する手法がある。このILPを向上させる手法としては、スーパ

スカラ アーキテクチャ (Super-scalar architecture) や、超長命令語アーキテクチャ (VLIW architecture: Very Long Instruction Word architecture) といった方式が知られている。

【0002】

一方、近年では、携帯電話、プリンタ、デジタルテレビといった機器にも、マイクロプロセッサが組み込まれている。このようにプロセッサが組み込まれた機器は組み込み機器 (Embedded Application Equipment) と呼ばれ、このように組み込まれたプロセッサは、組み込みプロセッサ (Embedded Processor) と呼ばれている。

【0003】

近年では、このような組み込み機器に求められる処理能力が、年々高くなってきていることから、組み込みプロセッサの処理能力も、年々高いものが求められるようになってきている。従って、組み込みプロセッサの実現方式として、ILPを向上させる手法を採用しているものもある。

【0004】

その一方で、組み込み機器の分野では、機器を安価に実現することと、機器の消費電力が低いことが求められている。よって、組み込みプロセッサとして、スーパースカラ・アーキテクチャを採用した場合、回路規模が大きくなり、その消費電力が高くなる。一方、超長命令語アーキテクチャを採用した場合、命令を配置しない命令スロットに対して、無操作命令 (NOP Instruction: No-operation Instruction) を配置する必要がある為、プログラム規模が大きくなり、大容量な記憶装置が必要とされるといった課題があった。

【0005】

このため、特願平11-281957 (平成11年10月1日出願) に示されるような可変長の超長命令語アーキテクチャを採用した組み込みプロセッサがある。

【0006】

本発明は、このようなプロセッサ向けの言語処理系において必要とされるVLIWにおける基本命令の並びを確認する技術に関するものである。

【0007】

【従来の技術】

図1に、超長命令語アーキテクチャに基づくプロセッサの構成を従来技術として示す。ここでは、このプロセッサをVLIWプロセッサと呼ぶ。

(構成)

図1に示される従来のプロセッサは、メモリ10、命令読出部11、命令レジスタ12、整数ユニット(IU: Integer Unit)であるIU0とIU1、浮動小数点数ユニット(FU: Floating Unit)であるFU0とFU1、分岐ユニット(BU: Branching Unit)であるBU0とBU1、汎用レジスタGR (General purpose Register)、浮動小数点レジスタFR (Floating Register)、プログラムカウンタPC (Program Counter)を備える。

【0008】

命令読出部11は、メモリ10上のPCが示すアドレスより1VLIW命令分のメモリ領域をメモリから読み出し、命令レジスタ12に書き込む。また、PCを1VLIW命令分だけインクリメントする。

【0009】

命令レジスタ12は、命令読出部11より書き込まれた命令を保持する。その命令をIU、FU、BUへ供給する。この供給は次の通りである。

【0010】

命令スロット0に保持されている基本命令はIU0へ供給される。命令スロット1に保持されている基本命令はFU0へ供給される。命令スロット2に保持されている基本命令はIU1へ供給される。命令スロット3に保持されている基本命令はFU2へ供給される。命令スロット4に保持されている基本命令はBU0へ供給される。命令スロット5に保持されている基本命令はBU1へ供給される。

【0011】

I U 0、I U 1 は、整数演算命令、整数ロード命令、整数ストア命令、浮動小数点数ロード命令、浮動小数点ストア命令、無操作命令を実行する。

【0012】

整数演算命令が供給された場合、入力オペランドデータを汎用レジスタ G R から読み出し、整数演算を行なった後、演算結果である出力オペランドデータを汎用レジスタ G R に書き込む。

【0013】

整数ロード命令が供給された場合、入力オペランドデータをレジスタから読み出し、実効アドレスを計算した後、メモリ上の実効アドレスに対応する領域を読み出し、その値を汎用レジスタ G R に書き込む。

【0014】

整数ストア命令が供給された場合、入力オペランドデータを汎用レジスタ G R から読み出し、実効アドレスを計算した後、メモリ上の実効アドレスに対応する領域へストアデータを書き込む。

【0015】

浮動小数点数ロード命令が供給された場合、入力オペランドデータをレジスタから読み出し、実効アドレスを計算した後、メモリ上の実効アドレスに対応する領域を読み出し、その値を浮動小数点レジスタ F R に書き込む。

【0016】

浮動小数点数ストア命令が供給された場合、入力オペランドデータを浮動小数点レジスタ F R から読み出し、実効アドレスを計算した後、メモリ上の実効アドレスに対応する領域へストアデータを書き込む。

【0017】

無操作命令が供給された場合、何も操作をしない。

【0018】

F U 0、F U 1 は、浮動小数点数演算命令、無操作命令を実行する。浮動小数点数演算命令が供給された場合、入力オペランドデータを浮動小数点レジスタ F R から読み出し、浮動小数点演算を行なった後、演算結果である出力オペランドデータを浮動小数点レジスタ F R に書き込む。無操作命令が供給された場合

、何も操作をしない。

【0019】

BU0、BU1は、無条件分岐命令、条件分岐命令、無操作命令を実行する。無条件分岐命令が供給された場合、入力オペランドデータをレジスタ（GR、PC）から読み出し、それらを用いてアドレス計算を行ない、その結果をプログラムカウンタPCに書き込む。条件分岐命令が供給された場合、分岐条件が成立するか否かを判定して、分岐が成立する場合には、入力オペランドデータをレジスタ（GR、PC）から読み出し、それらを用いてアドレス計算を行ない、その結果を分岐先の命令のアドレスとして、PCに書き込む。無操作命令が供給された場合、何も操作をしない。

【0020】

ここでは、IU、FU、BUを機能ユニットと呼ぶ。機能ユニットは、命令レジスタから供給された基本命令を実行する。

（動作）

次に、VLIWプロセッサの動作について説明する。

【0021】

図2のようなプログラムが与えられたときの、図1に示されるVLIWプロセッサの動作について、図3を用いて説明する。なお、図中において、ADDは加算を行なう整数演算命令、FADDは加算を行う浮動小数点数演算命令であり、NOPは無操作命令である。

（時刻1）

（A）命令読出部11では、PC内の命令アドレスを用いて、図2のようにメモリ10上に配置されたVLIW命令1をメモリ10から読み出し、命令レジスタ12に書き込む。このとき命令レジスタ12には、図3の時刻1のように格納される。

【0022】

（B）機能ユニットでは、供給された命令を実行する。命令スロット0のADDがIU0で実行される。命令スロット1のFADDがFU0で実行される。命令スロット2のADDがIU1で実行される。命令スロット3のFADDがFU

1で実行される。命令スロット4のNOPがBU0で実行される。命令スロット5のNOPがBU1で実行される。

【0023】

各機能ユニットにおいて、基本命令の実行が完了することで、VLIW命令1の実行が完了する。

(時刻2)

(A) 命令読出部11では、PC内の命令アドレスを用いて、図2のようにメモリ10上に配置されたVLIW命令2をメモリ10から読み出し、命令レジスタ12に書き込む。このとき命令レジスタ12には、図3の時刻2のように格納される。

【0024】

(B) 機能ユニットでは、供給された命令を実行する。命令スロット0のADDがIU0で実行される。命令スロット1のNOPがFU0で実行される。命令スロット2のNOPがIU1で実行される。命令スロット3のNOPがFU1で実行される。命令スロット4のNOPがBU0で実行される。命令スロット5のNOPがBU1で実行される。

【0025】

各機能ユニットにおいて、基本命令の実行が完了することで、VLIW命令2の実行が完了する。

(時刻3)

(A) 命令読出部11では、PC内の命令アドレスを用いて、図2のようにメモリ10上に配置されたVLIW命令3をメモリ10から読み出し、命令レジスタ12に書き込む。このとき命令レジスタ12には、図3の時刻3のように格納される。

【0026】

(B) 機能ユニットでは、供給された命令を実行する。命令スロット0のNOPがIU0で実行される。命令スロット1のFADDがFU0で実行される。命令スロット2のNOPがIU1で実行される。命令スロット3のNOPがFU1で実行される。命令スロット4のNOPがBU0で実行される。命令スロット5

のNOPがBU1で実行される。

【0027】

各機能ユニットにおいて、基本命令の実行が完了することで、VLIW命令3の実行が完了する。

【0028】

VLIWプロセッサでは、命令読出部11で読み出された1VLIW命令内の命令スロットの位置と、命令実行部で命令スロットが実行される機能ユニットとの関係は、1対1に定まる。すなわち、整数演算命令／整数ロード命令／整数ストア命令／浮動小数点数ロード命令、浮動小数点数ストア命令は、整数ユニットであるIU0、IU1のみで実行されるためにVLIW命令内部の命令スロット0か命令スロット2に配置される必要がある。

【0029】

浮動小数点数演算命令は浮動小数点数ユニットであるFU0、FU1のみで実行されるために、VLIW命令内部の命令スロット1か命令スロット3に配置される必要がある。

【0030】

条件分岐命令、無条件分岐命令は、分岐ユニットであるBU0、BU1のみで実行されるためにVLIW命令内部の命令スロット4か命令スロット5に配置される必要がある。そのため、VLIWプロセッサの言語処理系では、VLIW命令を構成する基本命令と命令スロット番号の対応を確認する必要がある。このため、VLIWプロセッサの言語処理系であるアセンブラ及びコンパイラにVLIW確認部を設け、VLIW命令が実行可能な基本命令の並びで構成されているかどうかを確認する。そして、確認できたVLIW命令のみをメモリ10に書き込む。

(アセンブラ)

図4は、VLIWプロセッサの言語処理系であるアセンブラの従来例を示すフローチャートである。アセンブラは、図4のフローチャートに示すように、字句解析部S11、命令コード生成部S12、VLIW確認部S13、及びオブジェクト生成部S14からなる。

【0031】

字句解析部 S 1 1 は、アセンブリ言語プログラムのソースコードのファイルから、テキストを前から順に読み込み、読み込んだソースコードのテキストの字句を解析する。命令コード生成部 S 1 2 は、解析した字句を命令コードに変換する。V L I W 確認部 S 1 3 は、V L I W 命令がプロセッサの命令発行部で、命令実行部に発行可能か否かを確認する。オブジェクト生成部 S 1 4 は、発行可能な V L I W 命令をオブジェクト形式に変換して、生成したオブジェクトをオブジェクトプログラムファイルに出力する。

【0032】

図 5 に、V L I W 確認部 S 1 3 のフローチャートを示す。V L I W 確認部 S 1 3 は、V L I W 命令の命令スロット配置確認部 S 1 3 - 1 と同一レジスタ書き込み確認部 S 1 3 - 2 から構成される。

【0033】

V L I W 命令の命令スロット配置確認部 S 1 3 - 1 では、V L I W 命令の各基本命令について配置可能な命令スロットに配置されているか否かを確認する。図 6 に、V L I W 命令の命令スロット配置確認部 S 1 3 - 1 のフローチャートを示す。

【0034】

同一レジスタ書き込み確認部 S 1 3 - 2 では、1 つの V L I W 命令内の基本命令が、同時に同一レジスタに書き込むか否かを確認する。V L I W 命令の命令スロット配置確認部 S 1 3 - 1 において、注目する V L I W 命令の基本命令が発行可能か否かを判定するアルゴリズムは、以下のとおりである。

【0035】

まず、ステップ S 2 2 で、注目する V L I W 命令より、基本命令を取り出す。次に、ステップ S 2 3 で、取り出した基本命令を配置している命令スロットを求める。次に、ステップ S 2 4 で、配置可能命令スロットテーブルから取り出した基本命令を配置可能な命令スロットを求める。そして、ステップ S 2 5 で、取り出した基本命令の命令スロットが配置可能な命令スロットとなっているか否かを確認する。上記ステップ S 2 2 - S 2 7 を繰り返して、全ての命令スロットを処

理する（ステップ S 2 1）。

【 0 0 3 6 】

図 7 に、ステップ S 2 4 で参照される配置可能命令スロットテーブルの構成を示す。配置可能命令スロットテーブルとは、V L I W プロセッサが有する各基本命令に対して、各命令スロット配置可能な否かの情報を保持する。

（コンパイラ）

図 7 は、V L I W プロセッサの言語処理系であるコンパイラの従来例を示すフローチャートである。図示するフローチャートに示すように、コンパイラは字句解析部 S 3 1、構文解析部 S 3 2、意味解析部 S 3 3、V L I W 形成部 S 3 4、及びアセンブリ言語記述出力部 S 3 5 からなる。

【 0 0 3 7 】

字句解析部 S 3 1 は、高級言語プログラムのソースコードのファイルから、テキストを前から順に読み込み、読み込んだソースコードのテキストの字句を解析する。構文解析部 S 3 2 は、プログラムの論理的構造の解明や構文規則との対応づけを行なう。意味解析部 S 3 3 では、各構成要素の意味の明確化と中間語への変換を行なう。V L I W 形成部 S 3 4 は中間語を V L I W 命令に変換するもので、コンパイラの V L I W 確認部として機能する。アセンブリ言語記述出力部 S 3 5 は、V L I W 命令のアセンブリ言語記述を出力する。

【 0 0 3 8 】

図 9 は、コンパイラの V L I W 形成部 S 3 4 のフローチャートを示す。V L I W 形成部 3 4 の V L I W 確認アルゴリズムは、以下のとおりである。まず、ステップ S 4 1 で、中間語表現から 1 基本命令を取り出せるかどうか判定する。可能であればステップ S 4 2 に行き、不可能であれば、S 4 8 に行く。ステップ S 4 2 で、中間言語から 1 基本命令を取り出す。ステップ S 4 3 で、基本命令を命令配置テーブルに配置できるか確認する。可能であれば、ステップ S 4 5 に行き、不可能であれば、ステップ S 4 6 に行く。

【 0 0 3 9 】

ステップ S 4 5 で、命令配置テーブルに基本命令を配置し、ステップ S 4 1 に戻る。ステップ S 4 4 で N O の場合、ステップ S 4 6 で命令配置テーブルに配置

された基本命令のセットをV L I W命令として出力する。ステップs 4 7で、命令配置テーブルのすべての配置済みエントリを未配置状態にし、ステップS 4 3に戻る。

【0 0 4 0】

ステップS 4 1がN Oの場合、命令配置テーブルに配置された基本命令のセットをV L I W命令として出力して終了する（ステップS 4 8）。

【0 0 4 1】

【発明が解決しようとする課題】

前述した特願平1 1 - 2 8 1 9 5 7 (平成1 1年1 0月1日出願)に示されるような可変長の超長命令語アーキテクチャを採用した組み込みプロセッサでは、V L I W命令の構成単位となる命令スロットと機能ユニットの対応関係が、1 対多、あるいは、多対多となる。よって、言語処理系では、V L I W命令における（基本）命令の組み合わせがプロセッサにおいて実行可能な組み合わせとなっているか否かを確認する必要がある。

【0 0 4 2】

また、組み込み機器の分野は、その適用分野が広範囲であることから、処理性能が比較的低いものから処理性能が非常に高いものまで、要求される処理性能の幅が広い。これに対して、先の特許出願に示されるような可変長の超長命令語アーキテクチャを採用し、命令発行幅が異なるプロセッサとして実現することが考えられる。これにより、処理性能が比較的低いものには、命令発行幅が小さいプロセッサを、処理性能が非常に高いものには、命令発行幅が大きいプロセッサを適用することが可能となる。しかしながら、各々のプロセッサ向けに、言語処理系を個別に用意するのは不経済である。

【0 0 4 3】

従って、本発明は、このような可変長の超長命令語アーキテクチャに基づいて命令発行幅が異なるプロセッサの言語処理系に適した命令処理方法及びこの処理をコンピュータに実行させるためのプログラムを提供することを目的とする。

【0 0 4 4】

【課題を解決するための手段】

本発明は、プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第1のステップと、基本命令を論理命令スロットに配置可能かどうかを判断する第2のステップと、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第3のステップとを有する命令処理方法である。

【0045】

論理命令スロットは、機能ユニット毎に対応して命令スロット（プロセッサが有する物理的な命令スロットで、以下の説明では物理命令スロットと言う場合もある）が存在すると考えた仮想的な命令スロットである。この論理命令スロットに配置可能かどうかを判断することで命令の長さをプロセッサが有する物理的な命令スロットに配置可能かどうかを判断できる。そして、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮するので、基本命令の並びに関する制約を満たす基本命令の並びが得られる。ただし、同一の命令種別において、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮する必要がある場合には、物理的な命令に配置可能と判断された基本命令を配置すればよい。

【0046】

また、本発明は、プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第1のステップと、当該命令カテゴリから実行可能な機能ユニットの種別を特定する第2のステップと、基本命令を論理命令スロットに配置可能かどうかを判断する第3のステップと、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの物理スロットに配置する第4のステップとを有する命令処理方法である。命令カテゴリは命令の分類を表わす。つまり、同一の命令種別の命令も複数カテゴリある（例えば、ある整数演算命令は整数演算を処理できるいずれの機能ユニットでも処理可能であるが、別の整数演算命令は特定の機能ユニットでしか処理できない）。従って、命令カテゴリに応じて実行可能な機能ユニットを特定することができるので、上記命令カテゴリに応じた基本

命令の配列を得ることができる。この場合にも、必要に応じて、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮する。

【発明の実施の形態】

本発明の説明の都合上、可変長命令を実行するプロセッサの一例として、可変長VLIWアーキテクチャに基づくプロセッサである可変長VLIWプロセッサの第1の構成例を最初に説明し、次に可変長VLIWプロセッサの言語処理系に適した本発明の命令処理方法の実施の形態について説明し、その後、可変長VLIWプロセッサの別の構成例について説明する。

（可変長VLIWプロセッサの第1の構成例）

図13に、最大4命令を同時に実行できる可変長VLIWプロセッサの第1の構成例を示す。このプロセッサは、メモリ20、命令読出部21、命令レジスタ22、命令発行部23、及び命令実行部24を有する。命令実行部24は、整数ユニットであるIU0とIU1、浮動小数点数ユニットであるFU0とFU1、分岐ユニットであるBU0とBU1を備える。また、プロセッサは汎用レジスタGR、浮動小数点レジスタFR、プログラムカウンタPCを有する。

【0047】

ここで、VLIW命令内の基本命令の並びとして、図11に示す22通りのVLIW命令を実行可能であるとする。図11において、記号の意味は次の通りである。I0は、IU0にて実行される基本命令が配置されることを意味する。I1は、IU1にて実行される基本命令が配置されることを意味する。F0は、FU0にて実行される基本命令が配置されることを意味する。F1は、FU1にて実行される基本命令が配置されることを意味する。B0は、BU0にて実行される基本命令が配置されることを意味する。B1は、BU1にて実行される基本命令が配置されることを意味する。空欄は、基本命令を配置しないことを意味する。

【0048】

命令読出部21は、メモリ20上のプログラムカウンタPCが示すアドレスより1VLIW命令分のメモリ領域をメモリ20から読み出し、命令レジスタ22

に書き込む。また、プログラムカウンタPCを1VLIW命令分だけインクリメントする。命令レジスタ22は、命令読出部21より書き込まれた命令を保持する。命令発行部23は、命令レジスタ22から読み込んだ命令を対応する機能ユニットであるIU、FU、BUへ供給する。最大4命令を同時実行可能で、6つの機能ユニットうち最大4つの機能ユニットに命令を供給する。各命令スロットに保持されている基本命令は以下のように機能ユニットに供給される。

【0049】

命令スロット0に保持されている基本命令はIU0、FU0、BU0へ供給可能である。命令スロット1に保持されている基本命令はFU0、IU1、FU1、BU0、BU1へ供給可能である。命令スロット2に保持されている基本命令はIU1、FU1、BU0、BU1へ供給可能である。命令スロット3に保持されている基本命令はFU1、BU0、BU1へ供給可能である。また、このプロセッサで許されているVLIW命令内の基本命令の並びは、図11の通りとする。

【0050】

IU0、IU1は、整数演算命令、整数ロード命令、整数ストア命令、浮動小数点数ロード命令、浮動小数点ストア命令を実行する。

【0051】

整数演算命令が供給された場合、入力オペランドデータを汎用レジスタGRから読み出し、整数演算を行なった後、演算結果である出力オペランドデータを汎用レジスタGRに書き込む。

【0052】

整数ロード命令が供給された場合、入力オペランドデータをレジスタから読み出し、実効アドレスを計算した後、メモリ20上の実効アドレスに対応する領域を読み出し、その値を汎用レジスタGRに書き込む。

【0053】

整数ストア命令が供給された場合、入力オペランドデータを汎用レジスタGRから読み出し、実効アドレスを計算した後、メモリ20上の実効アドレスに対応する領域へストアデータを書き込む。

【0054】

浮動小数点数ロード命令が供給された場合、入力オペランドデータをレジスタから読み出し、実効アドレスを計算した後、メモリ20上の実効アドレスに対応する領域を読み出し、その値を浮動小数点レジスタFRに書き込む。

【0055】

浮動小数点数ストア命令が供給された場合、入力オペランドデータを浮動小数点レジスタFRから読み出し、実効アドレスを計算した後、メモリ20上の実効アドレスに対応する領域へストアデータを書き込む。

【0056】

FU0、FU1は、浮動小数点数演算命令を実行する。

【0057】

浮動小数点数演算命令が供給された場合、入力オペランドデータを浮動小数点レジスタFRから読み出し、浮動小数点演算演算を行なった後、演算結果である出力オペランドデータを浮動小数点レジスタFRに書き込む。

【0058】

BU0、BU1は、無条件分岐命令、条件分岐命令を実行する。無条件分岐命令が供給された場合、入力オペランドデータをレジスタGR、PCから読み出し、それらを用いてアドレス計算を行ない、その結果をプログラムカウンタPCに書き込む。

【0059】

条件分岐命令が供給された場合、分岐条件が成立するか否かを判定して、分岐が成立する場合には、入力オペランドデータをレジスタGR、PCから読み出し、それらを用いてアドレス計算を行ない、その結果を分岐先の命令のアドレスとして、プログラムカウンタPCに書き込む。

【0060】

前述したように、IU、FU、BUを機能ユニットである。機能ユニットは、命令レジスタから供給された基本命令を実行する。また、可変長VLIWプロセッサでは、VLIW命令は図12のようにメモリ20上に配置される。VLIW命令を構成する各命令にはパッキングフラグが設けられており、VLIW命令中

の最後の命令のパッキングフラグはONとなっている。パッキングフラグを用いることで、V L I W命令の終り（次のV L I W命令の始まり）を特定することができる。

（可変長V L I Wプロセッサの第1の構成例の動作）

次に、第1の構成例の可変長V L I Wプロセッサの動作を説明する。

【0061】

図12のようなプログラムが与えられたときの図10に示されるV L I Wプロセッサの動作について、図13を用いて説明する。なお、図中において、ADDは加算を行なう整数演算命令、FADDは加算を行う浮動小数点数演算命令で。

（時刻1）

（A）命令読出部21は、プログラムカウンタPC内の命令アドレスを用いて、図12のようにメモリ上に配置されたV L I W命令1をメモリ20から読み出し、命令レジスタ22に書き込む。このとき命令レジスタには図13の時刻1のように格納される。

【0062】

（B）機能ユニットは、供給された命令を次の通り実行する。命令スロット0のADDがIU0で実行される。命令スロット1のFADDがFU0で実行される。命令スロット2のADDがIU1で実行される。命令スロット3のFADDがFU1で実行される。BU0、BU1は基本命令が供給されないため動作しない。

【0063】

IU0、IU1、FU0、FU1において、基本命令の実行が完了することで、V L I W命令1の実行が完了する。

（時刻2）

（A）命令読出部21は、プログラムカウンタPC内の命令アドレスを用いて、図12のようにメモリ20上に配置されたV L I W命令2をメモリから読み出し、命令レジスタ22に書き込む。このとき命令レジスタ22には図13の時刻2のように格納される。

【0064】

(B) 機能ユニットは、供給された命令を次の通り実行する。命令スロット0のADDがIU0で実行される。IU1、FU0、FU1、BU0、BU1は基本命令が供給されないため動作しない。

【0065】

IU0において、基本命令の実行が完了することで、VLIW命令2の実行が完了する。

(時刻3)

(A) 命令読出部21は、プログラムカウンタPC内の命令アドレスを用いて、図12のようにメモリ上に配置されたVLIW命令3をメモリから読み出し、命令レジスタ22に書き込む。このとき命令レジスタ22には図13のように格納される。

【0066】

(B) 機能ユニットは、供給された命令を次の通り実行する。命令スロット0のFADDがFU0で実行される。IU0、IU1、FU1、BU0、BU1は基本命令が供給されないため動作しない。

【0067】

FU0において、基本命令の実行が完了することで、VLIW命令3の実行が完了する。

【0068】

ここで、VLIW命令における基本命令の並びは、可変長VLIWプロセッサにおいて実行可能でなければならない。従って、可変長VLIW命令内を構成する基本命令の並びの条件を満たすか否かをアセンブラ及びコンパイラで判断する必要がある。

(可変長VLIWプロセッサのアセンブラ)

図14に、可変長VLIWプロセッサのアセンブラのフローチャートを示す。VLIWプロセッサのアセンブラの構造と、可変長VLIWプロセッサのアセンブラの構造との違いは、可変長VLIW確認部(図14では参照番号S13Aで示す)である。

【0069】

図 1 5 に、可変長 V L I W 確認部のフローチャートを示す。図 5 に示す従来技術の V L I W プロセッサのアセンブラとの違いは、図 1 5 に示すアセンブラが可変長 V L I W 命令内基本命令並び確認部 1 3 A - 1 を有する点である。可変長 V L I W 命令内基本命令並び確認部 1 3 A - 1 は、基本命令の並びは発行可能かどうかを確認する。図 1 5 には、1 V L I W 命令内で、命令の組み合わせ制限に違反していないかどうかを確認する可変長 V L I W 命令の命令組み合わせ確認部 1 3 A - 2 が設けられているが、これは必要に応じて任意に設けられるものである。つまり、命令組み合わせ確認部 1 3 A - 2 は省略しても良い。図 1 5 に示す同一レジスタ書き込み確認部 S 1 3 A - 3 は、図 5 に示す同一レジスタ書き込み確認部 S 1 3 - 2 と同じである。すなわち、同一レジスタ書き込み確認部 S 1 3 A - 3 は、1 V L I W 命令内で、同一レジスタへの書き込み制御に違反していないかどうかを確認する。

【 0 0 7 0 】

なお、可変長 V L I W 確認部 S 1 3 A の可変長 V L I W 命令を構成する基本命令の並びを確認するアルゴリズムについては、後述する。

(可変長 V L I W プロセッサのコンパイラ)

図 1 6 に、可変長 V L I W プロセッサのコンパイラのフローチャートを示す。また、図 1 7 に、コンパイラの V L I W 形成部 S 3 4 A の基本命令を命令配置テーブルに配置できるか確認する部分のフローチャートを示す。図 8 に示す V L I W プロセッサのコンパイラの構造と、図 1 6 に示す可変長 V L I W プロセッサのコンパイラの構造との違いは、V L I W 形成部 3 4 (図 8) と V L I W 形成部 S 3 4 A (図 1 6) とは、基本命令を命令配置テーブルに配置できるかを確認するためのアルゴリズムが相違する。より特定すれば、図 1 0 のステップ S 4 3 のアルゴリズムと図 1 7 のステップ S 4 3 A のアルゴリズムとは異なる。そして、V L I W 形成部 S 3 4 A のアルゴリズムは、図 1 4 の可変長 V L I W 確認部 S 1 3 A、より特定すれば可変長 V L I W 命令内基本命令並び確認部 S 1 3 A - 1 と同じアルゴリズムである。なお、このアルゴリズムについては後述する。

(可変長 V L I W 命令の基本命令の並びを確認するアルゴリズム)

次に、可変長 V L I W 命令を構成する基本命令の並びを確認するアルゴリズム

を説明する。

【0071】

このアルゴリズムは可変長VLIW確認部S13A（図14）の可変長VLIW命令内基本命令並び確認部S13A-1及びVLIW形成部S34A（図16）で用いられるもので、大別して以下の4通りある。

【0072】

第1のアルゴリズムは、可変長VLIWプロセッサの言語処理系において、可変長VLIW命令を構成する基本命令を実行できる機能ユニットの種別を特定する第1のステップと、基本命令を論理命令スロットに配置可能かどうかを判断する第2のステップと、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第3のステップとを有する。第1のアルゴリズムは例えば、後述するように、命令種別テーブル、命令スロットテーブル、命令配置テーブル及び機能ユニットテーブルを用いる。第1のステップは、命令種別テーブルを参照して基本命令の命令オペコードから命令を実行可能な機能ユニットの種別を得る。第2のステップは、命令スロットテーブルを参照して、基本命令を論理命令スロットに配置可能かどうかを判断する。第3のステップは、命令配置テーブル及び機能ユニットテーブルを用いて命令を配置する。このようにして配置された基本命令は、VLIW命令内に正しい並びで配置されたものである。

【0073】

第2のアルゴリズムは第1のアルゴリズムを変形したものであり、可変長VLIWプロセッサの言語処理系において、可変長VLIW命令を構成する基本命令の命令カテゴリを特定する第1のステップと、当該命令カテゴリから実行可能な機能ユニットの種別を特定する第2のステップと、基本命令を論理命令スロットに配置可能かどうかを判断する第3のステップと、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第4のステップとを有する。第2のアルゴリズムは例えば、後述するように、命令種別

テーブル、命令スロットテーブル、命令配置テーブル、機能ユニットテーブルに加え命令カテゴリテーブルを有する。第1のステップは、命令カテゴリテーブルを参照して、可変長VLIW命令を構成する基本命令の命令カテゴリを特定する。第2のステップは、命令種別テーブルを参照して、命令カテゴリから実行可能な機能ユニットの種別を特定する。第3のステップは、命令スロットテーブルを参照して、基本命令を論理命令スロットに配置可能かどうかを判断する。第4のステップは、命令配置テーブル及び機能ユニットテーブルを用いて命令を配置する。このようにして配置された基本命令は、VLIW命令内に正しい並びで配置されたものである。

【0074】

第3のアルゴリズムは、可変長VLIWプロセッサの言語処理系において、可変長VLIW命令を構成する基本命令を実行できる機能ユニットの種別を特定する第1のステップと、基本命令を論理命令スロットに配置可能かどうかを判断する第2のステップと、配置可能と判断された基本命令をプロセッサの命令スロットに配置する第3のステップとを有する。第3のアルゴリズムは、第1のアルゴリズムの第3のステップで行なっている論理命令スロットに配置される他の基本命令との関係を考慮することをしない。つまり、機能ユニットの同一種別内においてはどの機能ユニットにユニットに配置されていても良いという配置の条件がある場合に好適である。後述するように、第3のアルゴリズムは例えば、命令種別テーブル、命令スロットテーブル、命令配置テーブルを用いる。第1のステップは、命令種別テーブルを参照して基本命令の命令オペコードから命令を実行可能な機能ユニットの種別を得る。第2のステップは、命令スロットテーブルを参照して、基本命令を論理命令スロットに配置可能かどうかを判断する。第3のステップは、命令配置テーブル及び機能ユニットテーブルを用いて命令を配置する。このようにして配置された基本命令は、VLIW命令内に正しい並びで配置されたものである。

【0075】

第4のアルゴリズムは第3のアルゴリズムを変形したものであり、可変長VLIWプロセッサの言語処理系において、可変長VLIW命令を構成する基本命令

の命令カテゴリを特定する第1のステップと、当該命令カテゴリから実行可能な機能ユニットの種別を特定する第2のステップと、基本命令を論理命令スロットに配置可能かどうかを判断する第3のステップと、配置可能と判断された基本命令をプロセッサの命令スロットに配置する配置する第4のステップとを有する。第4のアルゴリズムは例えば、後述するように、命令種別テーブル、命令スロットテーブル、命令配置テーブル、機能ユニットテーブルに加え命令カテゴリテーブルを有する。第1のステップは、命令カテゴリテーブルを参照して、可変長VLIW命令を構成する基本命令の命令カテゴリを特定する。第2のステップは、命令種別テーブルを参照して、命令カテゴリから実行可能な機能ユニットの種別を特定する。第3のステップは、命令スロットテーブルを参照して、基本命令を論理命令スロットに配置可能かどうかを判断する。第4のステップは、命令配置テーブル及び機能ユニットテーブルを用いて命令を配置する。このようにして配置された基本命令は、VLIW命令内に正しい並びで配置されたものである。

【0076】

以下、第1～第4のアルゴリズムを順に説明する。

(第1の基本命令並び確認アルゴリズム)

第1の基本命令並び確認アルゴリズムは命令種別テーブル、命令スロットテーブル、命令配置テーブル、機能ユニットテーブルを用い、更に論理命令スロットポインタと物理命令スロットポインタを用いる。換言すれば、第1の基本命令並び確認アルゴリズムは命令種別テーブル、命令スロットテーブル、命令配置テーブル、機能ユニットテーブル、論理命令スロットポインタ、及び物理命令スロットポインタを含むデータ構造を用いている。

【0077】

図18に、命令種別テーブルの一例を示す。命令種別テーブルは基本命令に対して、それを実行可能な機能ユニット種別の情報を保持する。

【0078】

図19に、命令スロットテーブルの一例を示す。命令スロットテーブルは、論理命令スロットに対して、配置可能な機能ユニット種別を保持する。ここで論理命令スロットとは、各機能ユニットに対応して命令スロットが存在すると考えた

仮想的な命令スロットを意味する。

【0079】

図20に、命令配置テーブルの一例を示す。命令配置テーブルは、各命令スロットに対して、基本命令を配置しているか否かの情報を保持する。ここで物理命令スロットとはVLIW命令内の各命令スロットを意味する。

【0080】

図21に、機能ユニットテーブルの一例を示す。機能ユニットテーブルは、各機能ユニットに対して、すでに配置されて発行される命令があるか否かを示す情報を保持する。

【0081】

論理命令スロットポインタは、命令スロットテーブルの論理命令スロット番号を保持する。物理命令スロットポインタは、命令配置テーブルにおいて配置済みの基本命令数を保持する。

【0082】

図22は、第1の基本命令並び確認アルゴリズムを示すフローチャートである。まず、ステップS50で物理命令スロットポインタを0に、論理命令スロットポインタを0に初期化するステップS52は、命令を実行可能な機能ユニットの種別を得るステップである。配置する命令を命令列から命令情報を取り出し、その命令情報の命令オペコードを用いて命令種別テーブルを参照して、命令を実行可能な機能ユニット種別を得る。

【0083】

ステップS60は、配置可能な最も若い論理命令スロットを求めるステップで、ステップS61～S63を含む。ステップS61は、論理命令スロットポインタが指す命令スロットテーブルの論理命令スロットの配置可能な機能ユニット欄を参照して、命令を実行可能な機能ユニット種別に対応する欄に○があれば命令は配置可能であり（Yes）、そうでなければ命令を配置不可とする（No）。ステップS61で命令が配置不可と判断した場合（No）、ステップS62で論理命令スロットポインタをインクリメントする。論理命令スロットポインタが最大値を超えたら（No）、エラー終了する。論理命令スロットポインタが最大値

を超えていない場合には（Y e s）、次の論理命令スロットに対してステップ S 6 1～S 6 3 を繰り返す。

【0084】

ステップ S 7 0 は、命令を配置するステップで、ステップ S 7 1～S 7 6 を含む。ステップ S 6 1 の判断結果が Y e s の場合、ステップ S 7 1 で論理スロットが発行可能な機能ユニットの番号より若い機能ユニットに命令が配置されていることを、機能ユニットテーブルを参照して確認する。配置されていなければエラー終了する。

【0085】

ステップ S 7 1 は、論理スロットが発行可能な機能ユニットの番号より若い機能ユニットに命令が配置されていることを条件（制約）とする場合に対応するものである。この条件とは例えば、機能ユニット I U 0 に命令が配置されていなければ、機能ユニット I U 1 に命令を配置してはならないというものである。ステップ S 7 1 で命令が配置可能と判断された場合、ステップ S 7 2 で命令情報を命令配置テーブルに配置する。更に、機能ユニットに命令を配置したこと示すフラグを機能ユニットテーブルに設定する。ステップ S 7 3 で、論理命令スロットポインタをインクリメントする。命令のパッキングフラグが立っているならば V L I W 命令の最後の基本命令なので（ステップ S 7 4 の判断は Y e s）、ステップ S 7 5 で正常終了する。つまり、ステップ S 7 5 で命令スロット配置可とする。ステップ S 7 4 の判断結果が N o の場合には、ステップ S 7 6 で物理命令スロットポインタをインクリメントする。

【0086】

以上説明したステップ S 5 2 から S 7 0 までの処理を、物理命令スロットポインタが最大値を超えない範囲で繰り返す（ステップ S 5 1 と S 5 4）。超えてしまったら、ステップ S 5 5 で V L I W 命令内の命令数オーバーとして処理を終了する。

【0087】

このようにして第 1 の基本命令並び確認アルゴリズムを実行することにより、メモリ 20 に格納される V L I W 命令は可変長 V L I W プロセッサが実行するこ

とができる実行可能な機能ユニット種別を有した組み合わせとなっている。

【 0 0 8 8 】

以上説明した第 1 の基本命令並び確認アルゴリズムを前述した図 1 3 に示す可変長 V L I W プロセッサに適用した場合の命令種別テーブルと、命令スロットテーブルをそれぞれ図 2 3 及び図 2 4 に示す。これらの図は、図 2 1 に示す機能ユニットテーブルを統合した状態のテーブルを示している。

(第 2 の基本命令並び確認アルゴリズム)

第 2 の基本命令並び確認アルゴリズムは、第 1 の基本命令並び確認アルゴリズムで確認できる実行可能な機能ユニット種別に加え、命令カテゴリに応じた実行可能な機能ユニット番号を確認できることを特徴とする。

【 0 0 8 9 】

第 2 の基本命令並び確認アルゴリズムで用いるデータ構造は、命令種別テーブル、命令カテゴリテーブル、命令スロットテーブル、命令配置テーブル、機能ユニットテーブル、論理命令スロットポインタ、及び物理命令スロットポインタである。第 1 の基本命令並び確認アルゴリズムで用いるテーブルに加え、命令カテゴリテーブルを有する。

【 0 0 9 0 】

命令種別テーブルは、基本命令に対する命令カテゴリを保持する。図 2 5 に命令種別テーブルの一例を示す。ここで命令カテゴリとは、命令の分類を表すものである。I _ 1 は、整数カテゴリ 1 であり ADD 命令を含む。I _ 2 は整数カテゴリ 2 であり D I V 命令を含む。F _ 1 は浮動小数点数カテゴリ 1 であり FADD 命令を含む。B _ 1 は、分岐カテゴリ 1 であり BRA 命令を含む。

【 0 0 9 1 】

命令カテゴリテーブルは、基本命令に対して、配置可能な機能ユニット種別と機能ユニット種別毎の機能ユニット番号のビットパターンを保持する。図 2 6 に、命令カテゴリテーブルの一例を示す。整数カテゴリ I _ 2 はプロセッサ上の制約から、機能ユニット番号 2 の機能ユニットでは処理できない。

【 0 0 9 2 】

命令スロットテーブルは、論理命令スロットに対する配置可能な機能ユニット

種別と機能ユニット番号のビットパターンを保持する。図27に、命令スロットテーブルの一例を示す。ここで論理命令スロットとは、各機能ユニットに対応して命令スロットが存在すると考えた仮想的な命令スロットを意味する。

【0093】

命令配置テーブルは、各命令スロットに対するVLIW命令を構成する基本命令の情報を保持するもので、図20に示す第1の基本命令並び確認アルゴリズムで用いられる命令配置テーブルと同一構成である。

【0094】

機能ユニットテーブルは、各機能ユニットに配置されている命令があるか否かを示す情報を保持するもので、図21に示す第1の基本命令並び確認アルゴリズムで用いられる機能ユニットテーブルと同一構成である。

【0095】

論理命令スロットポインタは、命令スロットテーブルの論理命令スロット番号を保持する。物理命令スロットポインタは、命令配置テーブルにおいて配置済みの基本命令数を保持する。

【0096】

図28は、第2の基本命令並び確認アルゴリズムを示すフローチャートである。前述した図22の第1の基本命令並び確認アルゴリズムとの相違点として、図22のステップS52がステップS52Aに代わり、ステップS60がS60Aに代わり、ステップS57が新たに追加されている。

【0097】

ステップS50で、物理命令スロットポインタを0に、論理命令スロットポインタを0に初期化する。

【0098】

ステップS52Aは、命令の命令カテゴリを得るステップである。配置する命令を命令列から命令情報を取り出し、その命令情報の命令オペコードを用いて命令カテゴリテーブルを参照して、命令カテゴリを得る。

【0099】

ステップS57は、命令カテゴリから実行可能な機能ユニットを得るステップ

である。ステップ S 5 1 で得た命令カテゴリを用いて命令種別テーブルを参照して、命令を実行可能な機能ユニット種別と機能ユニット種別毎の機能ユニット番号のビットパターンを得る。

【 0 1 0 0 】

ステップ S 6 0 A は、配置可能な最も若い論理命令スロットを求めるステップである。論理命令スロットポインタがさす命令スロットテーブルの論理命令スロットの配置可能な機能ユニット欄を参照して、命令を実行可能な機能ユニット種別に対応する欄に○があり、かつ、機能ユニット種別毎の機能ユニット番号のビットパターンのビット和をとり 0 でなければ命令を配置可能と判断し、そうでなければ、命令を配置不可とする。ステップ S 6 1 A で命令が配置不可と判断されれば、ステップ S 6 2 で論理命令スロットポインタをインクリメントする。命令が配置可能であれば、ステップ S 7 1 に行く。ステップ S 6 3 で論理命令スロットポインタが最大値を超えたら、エラー終了する。次の論理命令スロットに対してステップ S 6 1 A、S 6 2、S 6 3 を繰り返す。

【 0 1 0 1 】

ステップ S 7 0 は命令を配置するステップである。ステップ S 7 1 で、論理スロットが発行可能な機能ユニットの番号より若い機能ユニットに命令が配置されていることを、機能ユニットテーブルを参照して確認して、配置されていないければエラー終了する。ステップ S 7 1 の判断結果が Y e s の場合には命令が配置可能であるので、命令情報を命令配置テーブルに配置する。更に、機能ユニットに命令を配置したこと示すフラグを機能ユニットテーブルに設定する。次に、ステップ S 7 2 で論理命令スロットポインタをインクリメントする。ステップ S 7 4 で、命令のパッキングフラグが立っているならばこの命令は V L I W 命令の最後の命令なので、ステップ S 7 5 で正常終了する。最後の命令でないと判断された場合には、ステップ S 7 6 で物理命令スロットポインタをインクリメントする。

【 0 1 0 2 】

以上説明したステップ S 5 2 から S 7 0 までの処理を、物理命令スロットポインタが最大値を超えない範囲で繰り返す（ステップ S 5 1 と S 5 4）。超えてしまったら、ステップ S 5 5 で V L I W 命令内の命令数オーバーとして処理を終了

する。

【0103】

このようにして第2の基本命令並び確認アルゴリズムを実行することにより、メモリ20に格納されるVLIW命令は、可変長VLIWプロセッサが実行することができる実行可能な機能ユニット種別に加え、命令カテゴリに応じた実行可能な機能ユニット番号を有した組み合わせとなっている。

（第3の基本命令並び確認アルゴリズム）

第3の基本命令並び確認アルゴリズムは、第1の基本命令並び確認アルゴリズムで用いていた機能ユニットテーブルを持たないデータ構造を有する。つまり、第3の基本命令並び確認アルゴリズムで用いるデータ構造は、命令種別テーブル、命令スロットテーブル、命令配置テーブル、論理命令スロットポインタ、及び物理命令スロットポインタである。

【0104】

図29は、第2の基本命令並び確認アルゴリズムを示すフローチャートである。前述した図22の第1の基本命令並び確認アルゴリズムとの相違点として、図22のステップ70に代えてステップ70Aが用いられている。ステップS70Aは、ステップ70からステップS71を取り除いた構成である。つまり、第3の基本命令並び確認アルゴリズムは、論理スロットが発行可能な機能ユニットの番号より若い機能ユニットに命令が配置されていることを、機能ユニットテーブルを参照して確認する処理をしない。従って、例えば図11に示す基本命令の並びにおいて命令スロット1に配置されるI1のみで構成されるVLIW命令も正常な基本命令の並びであると判断する。

【0105】

このように第3の基本命令並び確認アルゴリズムは、ステップS71の制約を持たない基本命令の並びを確認するのに適している。

（第4の基本命令並び確認アルゴリズム）

第4の基本命令並び確認アルゴリズムは、第3の基本命令並び確認アルゴリズムで用いた命令種別テーブル、命令スロットテーブル及び命令配置テーブルに加え、命令カテゴリテーブルを有する。つまり、第4の基本命令並び確認アルゴリ

ズムで用いるデータ構造は、命令種別テーブル、命令カテゴリテーブル、命令スロットテーブル、命令配置テーブル、論理命令スロットポインタ、及び物理命令スロットポインタである。第4の基本命令並び確認アルゴリズムは第3の基本命令並び確認アルゴリズムと同様に、第1及び第2の基本命令並び確認アルゴリズムで用いる機能ユニットテーブルを持たない。

【0106】

図30は、第4の基本命令並び確認アルゴリズムを示すフローチャートである。前述した図29の第3の基本命令並び確認アルゴリズムとの相違点として、図29のステップS52がステップS52Aに代わり、ステップS60がS60Aに代わり、ステップS57が新たに追加されている。ステップS52A、S60A及びステップS57は図28を参照して説明した通りである。また、図28の第2の基本命令並び確認アルゴリズムとの相違点は、図28のステップS70に代えて図29に示すステップS70Aを用いる点である。

【0107】

このように第4の基本命令並び確認アルゴリズムは、ステップS71の制約を持たない命令の配列であって、かつ命令カテゴリに応じた基本命令の並びを確認するのに適している。

【0108】

上記第1から第4の基本命令並び確認アルゴリズムは、図12に示す可変長VLIWプロセッサの第1の構成例のみならず、様々な形態の可変長VLIWプロセッサに適用できる。以下、可変長VLIWプロセッサの別の構成例について説明する。

(可変長VLIWプロセッサの第2の構成例)

最大8命令同時実行可能な可変長VLIWプロセッサを第2の構成例として図31に示す。この構成例は、アセンブラにおける前述した第1の基本命令並び確認アルゴリズムを適用してVLIW命令における基本命令の並びを確認して得られたものである。

【0109】

図31において、可変長VLIWプロセッサはメモリ120、命令読出部12

1、命令レジスタ122、命令発行部123、命令実行部124、汎用レジスタGR、浮動小数点レジスタFR、及びプログラムカウンタPCを有する。命令実行部124は整数ユニットIU0、IU1、IU2及びIU3、浮動小数点数ユニットFU0、FU1、FU2及びFU3、及び分岐ユニットBU0、BU1、BU2及びBU3を有する。

【0110】

また、VLIW命令内の基本命令の並びは、図32及び図33に示す73通りのVLIWが実行可能であるとする。

【0111】

命令読出部121は、メモリ120上のPCが示すアドレスより1VLIW命令分のメモリ領域をメモリ120から読み出し、命令レジスタ122に書き込む。また、プログラムカウンタPCを1VLIW命令だけインクリメントする。命令レジスタ122は、命令読出部121より書き込まれた命令を保持する。命令発行部123は、命令レジスタ122から読み込んだ命令を対応する機能ユニットであるIU、FU、BUへ供給する。最大8命令を同時実行可能で、10つの機能ユニットうち最大8つの機能ユニットに命令を供給する。各命令スロットに保持されている基本命令は以下のように機能ユニットに供給される。

【0112】

命令スロット0に保持されている基本命令はIU0、FU0、BU0へ供給可能である。命令スロット1に保持されている基本命令はFU0、IU1、FU1、BU0、BU1へ供給可能である。命令スロット2に保持されている基本命令はIU1、FU1、IU2、FU2、BU0、BU1へ供給可能である。命令スロット3に保持されている基本命令はFU1、IU2、FU2、IU3、FU3、BU0、BU1へ供給可能である。命令スロット4に保持されている基本命令はIU2、FU2、IU3、FU3、BU0へ供給可能である。命令スロット5に保持されている基本命令はFU2、IU3、FU3、BU0、BU1へ供給可能である。命令スロット6に保持されている基本命令はIU3、FU3、BU0、BU1へ供給可能である。命令スロット7に保持されている基本命令はFU3、BU0、BU1へ供給可能である。

IU0、IU1、IU3、IU3は、整数演算命令、整数ロード命令、整数ストア命令、浮動小数点数ロード命令、浮動小数点数ストア命令を実行する。

【0113】

整数演算命令が供給された場合、入力オペランドデータを汎用レジスタGRから読み出し、整数演算を行なった後、演算結果である出力オペランドデータを汎用レジスタGRに書き込む。整数ロード命令が供給された場合、入力オペランドデータをレジスタから読み出し、実効アドレスを計算した後、メモリ120上の実効アドレスに対応する領域を読み出し、その値を汎用レジスタGRに書き込む。整数ストア命令が供給された場合、入力オペランドデータを汎用レジスタGRから読み出し、実効アドレスを計算した後、メモリ120上の実効アドレスに対応する領域へストアデータを書き込む。浮動小数点数ロード命令が供給された場合、入力オペランドデータをレジスタから読み出し、実効アドレスを計算した後、メモリ120上の実効アドレスに対応する領域を読み出し、その値を浮動小数点レジスタFRに書き込む。浮動小数点数ストア命令が供給された場合、入力オペランドデータをFRから読み出し実効アドレスを計算した後、メモリ120上の実効アドレスに対応する領域へストアデータを書き込む。

【0114】

FU0、FU1、FU2、FU3は、浮動小数点数演算命令を実行する。浮動小数点数演算命令が供給された場合、入力オペランドデータを浮動小数点レジスタFRから読み出し、浮動小数点演算を行なった後、演算結果である出力オペランドデータを浮動小数点レジスタFRに書き込む。

【0115】

BU0、BU1無条件分岐命令、条件分岐命令を実行する。無条件分岐命令が供給された場合、入力オペランドデータをレジスタGR、PCから読み出し、それらを用いてアドレス計算を行ない、その結果をプログラムカウンタPCに書き込む。条件分岐命令が供給された場合、分岐条件が成立するか否かを判定して、分岐が成立する場合には、入力オペランドデータをレジスタGR、PCから読み出し、それらを用いてアドレス計算を行ない、その結果を分岐先の命令のアドレスとして、プログラムカウンタPCに書き込む。

【 0 1 1 6 】

また、可変長 V L I W 命令を実行するプロセッサでは、V L I W 命令は図 3 4 のようにメモリ上に配置される。

(可変長 V L I W プロセッサの第 2 の構成例の動作)

図 3 4 に示すようなプログラムが与えられたときの、図 3 1 に示される V L I W プロセッサの動作について、図 3 5 を用いて説明する。なお、図中において、A D D は加算を行なう整数演算命令、F A D D は加算を行う浮動小数点数演算命令である。

(時刻 1)

(A) 命令読出部 1 2 1 では、プログラムカウンタ P C 内の命令アドレスを用いて、図 3 4 のようにメモリ 1 2 0 上に配置された V L I W 命令 1 をメモリから読み出し、命令レジスタ 1 2 2 に書き込む。このとき命令レジスタ 1 2 2 には図 3 5 の時刻 1 のように格納される。

【 0 1 1 7 】

(B) 機能ユニットでは、供給された命令を実行する。命令スロット 0 の A D d が I U 0 で実行される。命令スロット 1 の F A D D が F U 0 で実行される。命令スロット 2 の A D D が I U 1 で実行される。命令スロット 3 の F A D D が F U 1 で実行される。命令スロット 4 の A D D が I U 2 で実行される。命令スロット 5 の F A D D が F U 2 で実行される。命令スロット 6 の A D D が I U 3 で実行される。命令スロット 7 の F A D D が F U 3 で実行される。B U 0、B U 1 は基本命令が供給されないため動作しない。

【 0 1 1 8 】

I U 0、I U 1、I U 2、I U 3、F U 0、F U 1、F U 2、F U 3 において、基本命令の実行が完了することで、V L I W 命令 1 の実行が完了する。

【 0 1 1 9 】

最大 8 命令同時実行可能な上記可変長 V L I W プロセッサのアセンブラの基本命令並び確認部 S 1 3 A - 1 に、第 1 の基本命令並びアルゴリズムを用いて実現した場合の命令種別テーブルを図 3 6 に、命令スロットテーブルを図 3 7 示す。

(可変長 V L I W プロセッサの第 3 の構成例)

最大 4 命令同時実行可能な可変長 V L I W プロセッサに、前述した第 2 の基本命令並び確認アルゴリズムを適用して V L I W 命令における基本命令の並びを確認した場合を、第 3 の構成例として以下に示す。

【 0 1 2 0 】

プロセッサ構造は、第 1 の構成例のプロセッサ構造と同じである。可変長 V L I W の命令並び確認部 1 3 A - 1 のアルゴリズムは、図 2 8 に示す第 2 の基本命令並び確認アルゴリズムを用いている。

【 0 1 2 1 】

図 3 8 に、第 3 の構成例の命令種別テーブルを示す。図 3 9 に、第 3 の構成例の命令カテゴリテーブルを示す。図 4 0 に、第 3 の構成例の命令スロットテーブルを示す。

(可変長 V L I W プロセッサの第 4 の構成例)

最大 8 命令同時実行可能な可変長 V L I W プロセッサに、前述した第 2 の基本命令並び確認アルゴリズムを適用して V L I W 命令における基本命令の並びを確認した場合を、第 4 の構成例として以下に示す。

【 0 1 2 2 】

プロセッサ構造は、第 2 の構成例のプロセッサ構造と同じである。可変長 V L I W の命令並び確認部 1 3 A - 1 のアルゴリズムは、図 2 8 に示す第 2 の基本命令並び確認アルゴリズムを用いている。

【 0 1 2 3 】

図 4 1 に、第 4 の構成例の命令種別テーブルを示す。図 4 2 に、第 4 の構成例の命令カテゴリテーブルを示す。図 4 3 に、第 4 の構成例の命令スロットテーブルを示す。

【 0 1 2 4 】

以上、本発明の実施の形態を説明した。前述の第 1 ないし第 4 のアルゴリズムは、コンピュータで処理される。つまり、コンピュータが実行するプログラムに第 1 ないし第 4 のアルゴリズムを構成するステップを記述し、コンピュータにこれらのステップを実行させる。つまり、本発明はコンピュータプログラムも含む。また、本発明はこれらのプログラムを格納するコンピュータ読取可能な記録媒

体をも含む。更に、本発明は第 1 ないし第 4 のアルゴリズムを構成するステップを手段とする情報処理装置をも含む。

【 0 1 2 5 】

可変長 V L I W プロセッサは上述した構成・動作のプロセッサに限定されるものではなく、本発明の命令処理方法は他の構成・動作の可変長 V L I W プロセッサに適用できる。

【 0 1 2 6 】

最後に、本発明の特徴の一部を以下に整理して示す。

(付記 1) プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第 1 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 2 のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第 3 のステップと

を有する命令処理方法。

(付記 2) プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第 1 のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第 2 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 3 のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第 4 のステップと

を有する命令処理方法。

(付記 3) プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第 1 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 2 のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する第3のステップと

を有する命令処理方法。

(付記4) プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第1のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第2のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第3のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する配置する第4のステップと

を有する命令処理方法。

(付記5) 前記第2のステップは、配置可能な最も若い論理命令スロットを特定するステップを有する付記1又は3に記載の命令処理方法。

(付記6) 前記第3のステップは、配置可能な最も若い論理命令スロットを特定するステップを有する付記2又は4に記載の命令処理方法。

(付記7) 前記第1ないし第3のステップを、プロセッサの全ての命令スロットに対して行なう付記1又は3に記載の命令処理方法。

(付記8) 前記第1ないし第4のステップを、プロセッサの全ての命令スロットに対して行なう付記2又は4に記載の命令処理方法。

(付記9) コンピュータに、

プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第1のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第2のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第3のステップと

を実行させるためのプログラム。

(付記 1 0) コンピュータに、

プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第 1 のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第 2 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 3 のステップと

配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する配置する第 4 のステップと

を実行させるためのプログラム。

(付記 1 1) コンピュータに、

プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第 1 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 2 のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する第 3 のステップと

を実行させるためのプログラム。

(付記 1 2) コンピュータに、

プロセッサが実行する命令を構成する基本命令の命令カテゴリを特定する第 1 のステップと、

当該命令カテゴリから実行可能な機能ユニットの種別を特定する第 2 のステップと、

基本命令を論理命令スロットに配置可能かどうかを判断する第 3 のステップと

配置可能と判断された基本命令をプロセッサの命令スロットに配置する配置する第 4 のステップと

を実行させるためのプログラム。

(付記 1 3) 付記 1 ないし 4 のいずれか一項に記載の前記命令処理方法を含むプロセッサのアセンブラ。

(付記 1 4) 付記 1 ないし 4 のいずれか一項に記載の前記命令処理方法を含むプロセッサのコンパイラ。

【 0 1 2 7 】

【発明の効果】

以上説明したように、本発明によれば、可変長の超長命令語アーキテクチャに基づいて命令発行幅が異なるプロセッサの言語処理系に適した命令処理方法を提供することができる。

【図面の簡単な説明】

【図 1】

超長命令語アーキテクチャに基づくプロセッサの従来構成のブロック図である。

【図 2】

図 1 に示すメモリ上での V L I W 命令の配置例を示す図である。

【図 3】

図 1 に示す命令レジスタに V L I W 命令が格納される様子を示す図である。

【図 4】

V L I W プロセッサの従来のアセンブラのフローチャートである。

【図 5】

従来配置可能命令スロットテーブルの一例を示す図である。

【図 6】

従来のアセンブラの V L I W 命令の確認部を示す図である。

【図 7】

従来のアセンブラの V L I W 命令の命令スロットに配置可能かどうかを確認するフローチャートである。

【図 8】

従来 V L I W プロセッサのコンパイラのフローチャートである。

【図 9】

従来のコンパイラの V L I W 形成部のフローチャートである。

【図 1 0】

可変長 V L I W プロセッサの第 1 の構成例を示すブロック図である。

【図 1 1】

図 1 0 に示す可変長 V L I W プロセッサの実行可能な V L I W 内の基本命令の並びを示す図である。

【図 1 2】

図 1 0 に示すメモリ上での V L I W 命令の配置例を示す図である。

【図 1 3】

図 1 0 に示す命令レジスタに V L I W 命令が格納される様子の一例を示す図である。

【図 1 4】

本発明の一実施の形態による可変長 V L I W プロセッサのアセンブラのフローチャートを示す図である。

【図 1 5】

図 1 3 に示す可変長 V L I W 確認部の一構成例を示すフローチャートである。

【図 1 6】

本発明の一実施の形態による可変長 V L I W プロセッサのコンパイラのフローチャートを示す図である。

【図 1 7】

図 1 6 に示す V L I W 形成部の一構成例を示すフローチャートである。

【図 1 8】

命令格納テーブルの一例を示す図である。

【図 1 9】

命令スロットテーブルの一例を示す図である。

【図 2 0】

命令配置テーブルの一例を示す図である。

【図 2 1】

機能ユニットテーブルの一例を示す図である。

【図 2 2】

可変長 V L I W 命令を構成する基本命令の並びを確認する第 1 のアルゴリズムを示すフローチャートである。

【図 2 3】

図 1 6 に示す可変長形成部の第 1 の構成例における命令種別テーブルを示す図である。

【図 2 4】

図 1 6 に示す可変長形成部の第 1 の構成例における命令スロットテーブルを示す図である。

【図 2 5】

可変長 V L I W 命令を構成する基本命令の並びを確認する第 2 のアルゴリズムで用いられる命令種別テーブルの一例を示す図である。

【図 2 6】

可変長 V L I W 命令を構成する基本命令の並びを確認する第 2 のアルゴリズムで用いられる命令カテゴリテーブルの一例を示す図である。

【図 2 7】

可変長 V L I W 命令を構成する基本命令の並びを確認する第 2 のアルゴリズムで用いられる命令スロットテーブルの一例を示す図である。

【図 2 8】

可変長 V L I W 命令を構成する基本命令の並びを確認する第 2 のアルゴリズムを示すフローチャートである。

【図 2 9】

可変長 V L I W 命令を構成する基本命令の並びを確認する第 3 のアルゴリズムを示すフローチャートである。

【図 3 0】

可変長 V L I W 命令を構成する基本命令の並びを確認する第 4 のアルゴリズムを示すフローチャートである。

【図 3 1】

可変長 V L I W プロセッサの第 2 の構成例を示すブロック図である。

【図 3 2】

図 3 1 に示す可変長 V L I W プロセッサで実行可能な V L I W 内の基本命令の並びを示す図（その 1）である。

【図 3 3】

図 3 1 に示す可変長 V L I W プロセッサで実行可能な V L I W 内の基本命令の並びを示す図（その 2）である。

【図 3 4】

図 3 1 に示すメモリ上での V L I W 命令の配置例を示す図である。

【図 3 5】

図 3 1 に示す命令レジスタに V L I W 命令が格納される様子を示す図である。

【図 3 6】

図 3 1 に示す第 2 の構成例における命令種別テーブルを示す図である。

【図 3 7】

図 3 1 に示す第 2 の構成例における命令スロットテーブルを示す図である。

【図 3 8】

可変長 V L I W プロセッサの第 3 の構成例における命令種別テーブルを示す図である。

【図 3 9】

可変長 V L I W プロセッサの第 3 の構成例における命令カテゴリテーブルを示す図である。

【図 4 0】

可変長 V L I W プロセッサの第 3 の構成例における命令スロットテーブルを示す図である。

【図 4 1】

可変長 V L I W プロセッサの第 4 の構成例における命令種別テーブルを示す図である。

【図 4 2】

可変長 V L I W プロセッサの第 4 の構成例における命令カテゴリテーブルを示す図である。

す図である。

【図 4 3】

可変長 V L I W プロセッサの第 4 の構成例における命令スロットテーブルを示す図である。

【符号の説明】

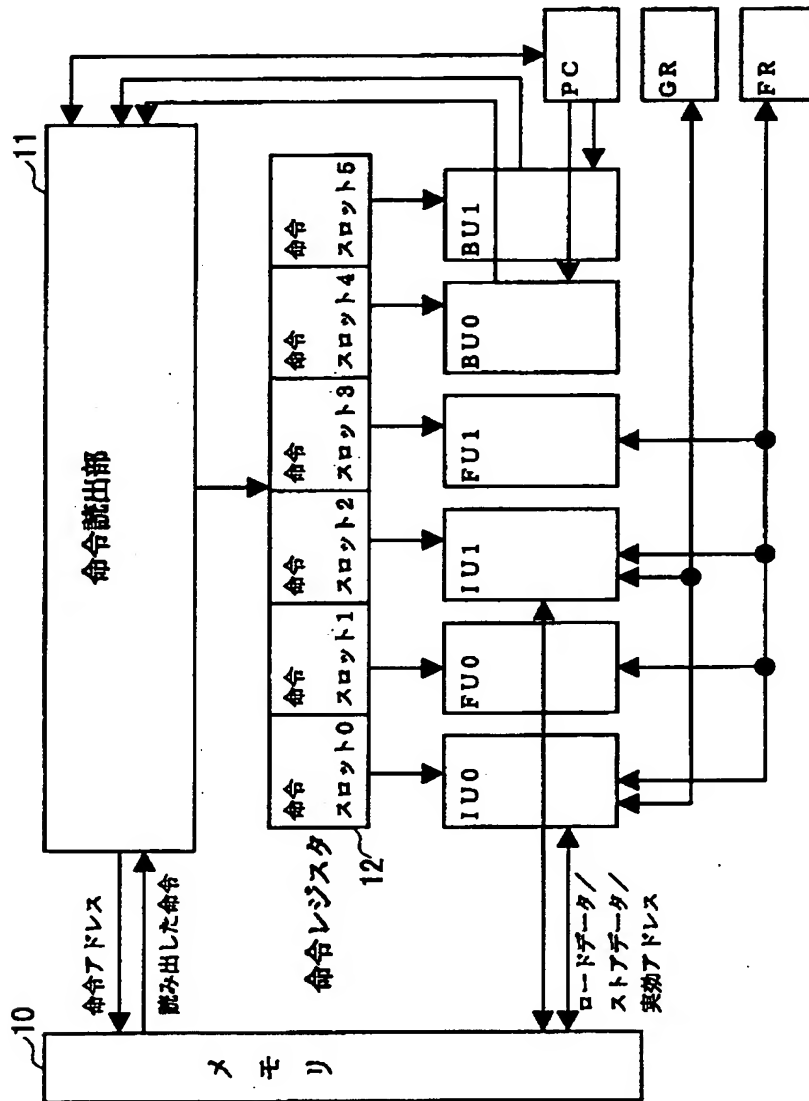
2 0	メモリ
2 1	命令読出部
2 2	命令レジスタ
2 3	命令発行部
2 4	命令実行部
P C	プログラムカウンタ
G R	汎用レジスタ
F R	浮動小数点レジスタ

【書類名】

図面

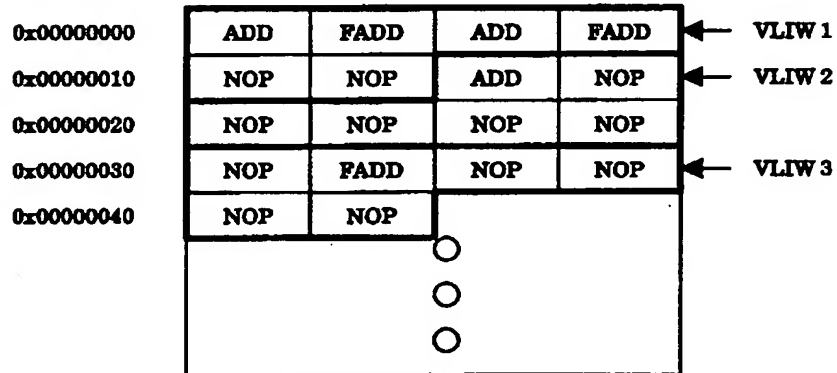
【図 1】

超長命令語アーキテクチャに基づくプロセッサの
従来構成のブロック図



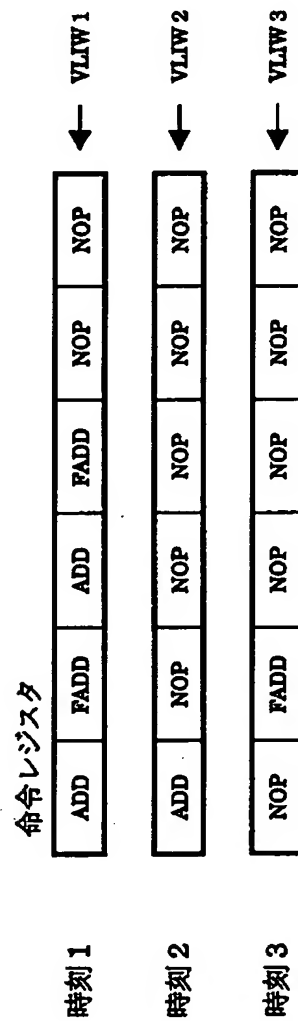
【図 2】

図 1 に示すメモリ上での VLIW 命令の配置例を示す図



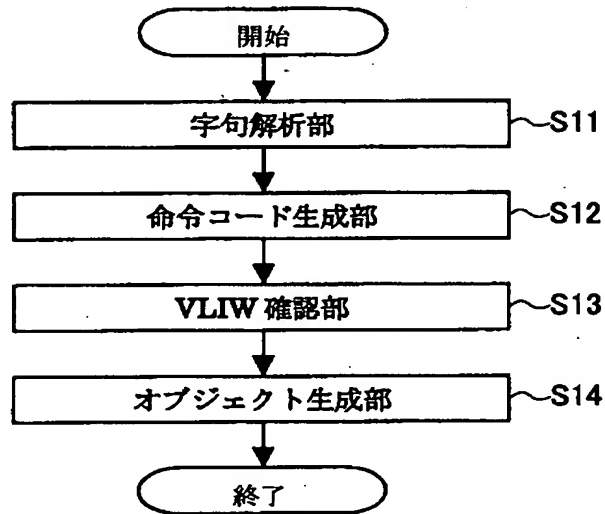
【図 3】

図 1 に示す命令レジスタに VLIW 命令が格納される様子を示す図



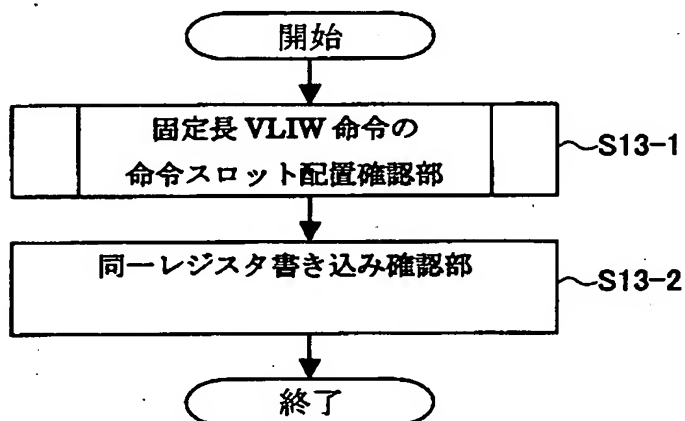
【図 4】

VLIWプロセッサの従来のアセンブラのフローチャート



【図 5】

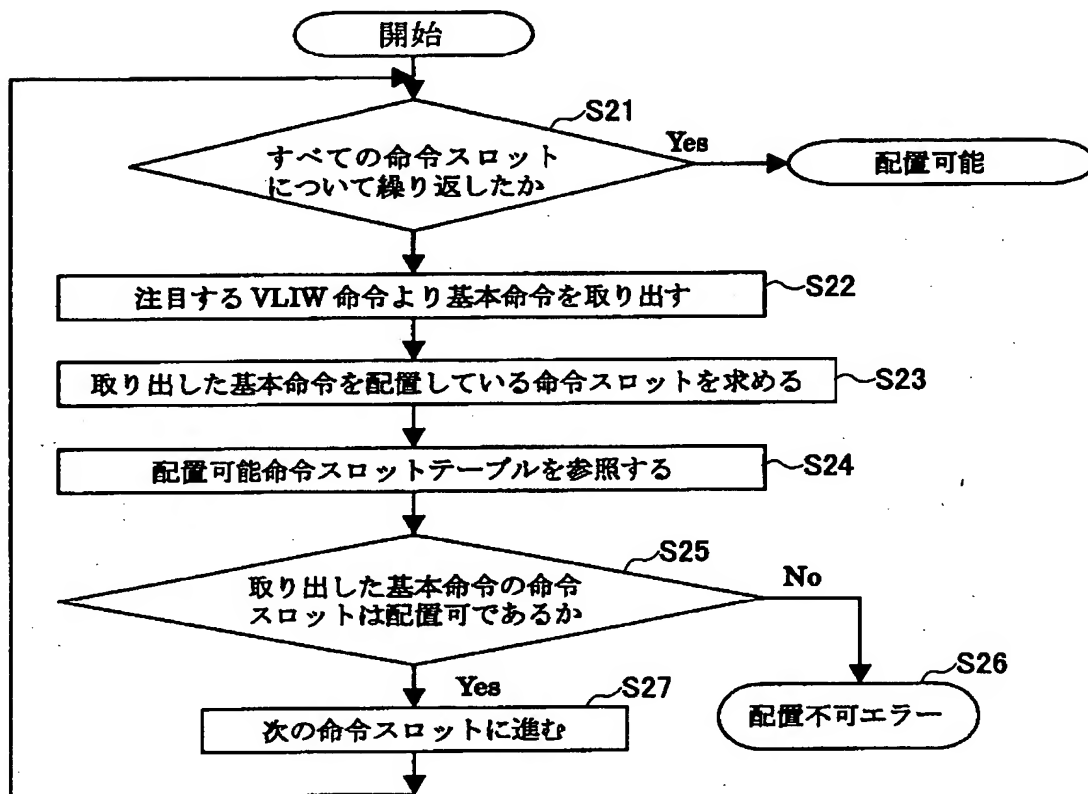
従来の配置可能命令スロットテーブルの一例を示す図



【図 6】

従来のアセンブラのVLIW命令の確認部を示す図

S13-1



【図 7】

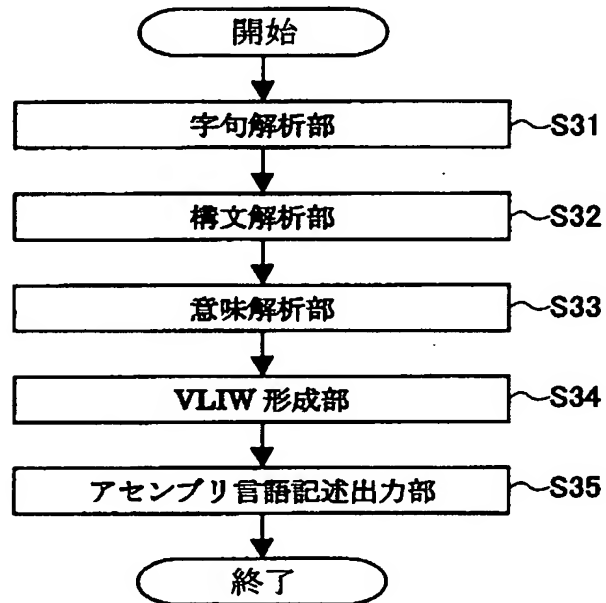
従来のアセンブラのVLW命令の命令スロットに
配置可能かどうかを確認するフローチャート

基本命令	命令スロット					
	スロット0	スロット1	スロット2	スロット3	スロット4	スロット5
ADD	○	—	○	—	—	—
FADD	—	○	—	○	—	—
BRA	—	—	—	—	○	○

○：配置可
—：配置不可

【図 8】

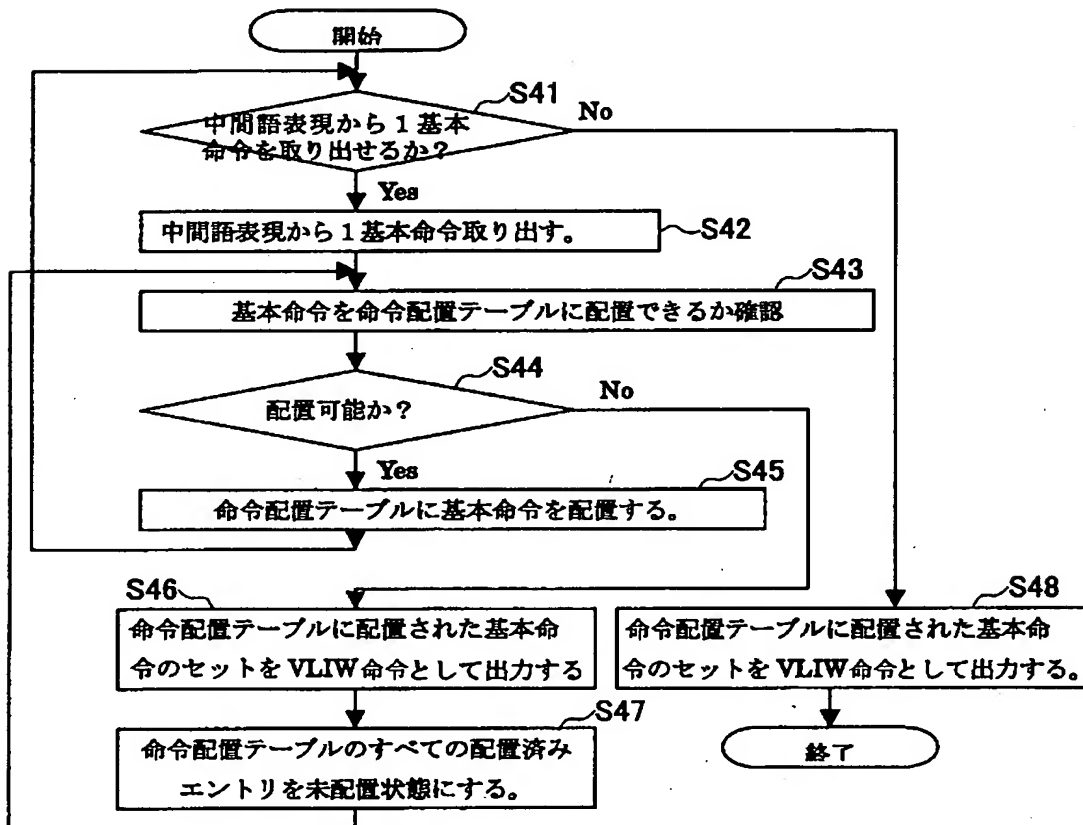
従来のVLIWプロセッサのコンパイラのフローチャート



【図 9】

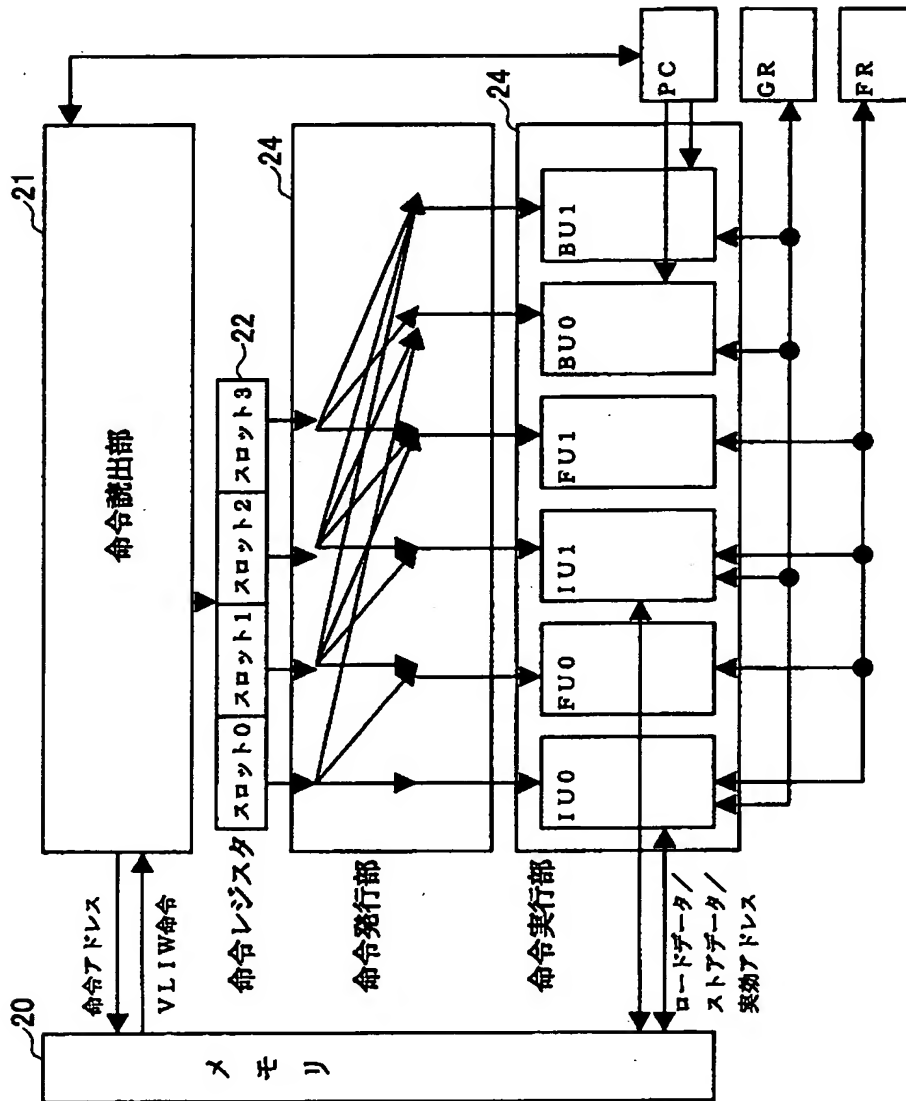
従来のコンパイラのVLIW形成部のフローチャート

S34



【図10】

可変長VLIWプロセッサの第1の構成例を示すブロック図



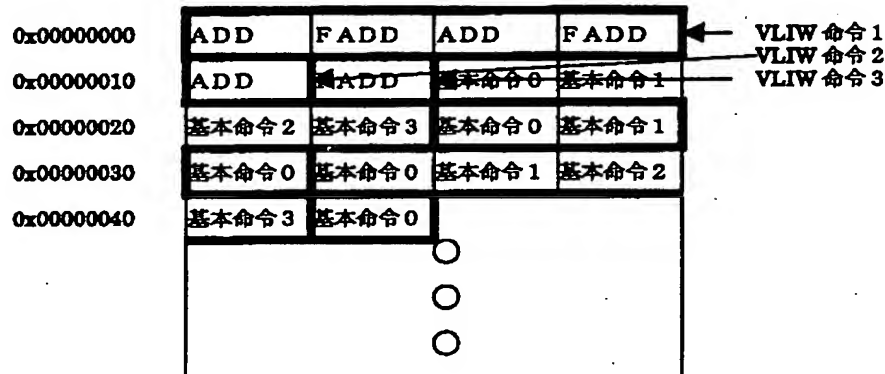
【図 11】

図10に示す可変長VLIWプロセッサの実行可能な
VLIW内の基本命令の並びを示す図

命令スロ ット0	命令スロ ット1	命令スロ ット2	命令スロ ット3
I0	F0	I1	F0
I0	F0	I1	B0
I0	F0	I1	
I0	F0	I1	B0
I0	F0	F1	
I0	F0	F1	B0
I0	F0	B0	B1
I0	F0	B	
I0	F0		
I0	I1	B0	B1
I0	I1	B0	
I0	I1		
F0	F1	B0	B1
F0	F1	B0	
F0	F1		
I0	B0	B1	
I0	B0		
I0			
F0	B0	B1	
F0	B0		
F0			
B0	B1		
B0			

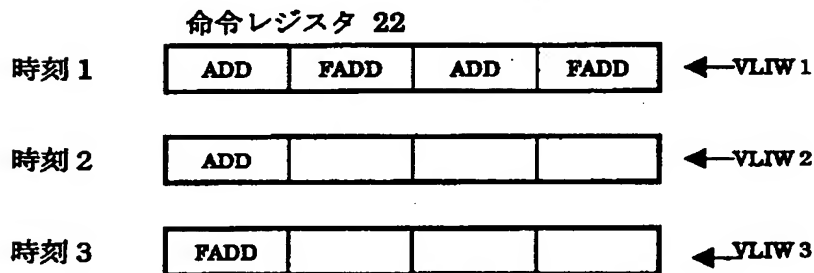
【図 12】

図10に示すメモリ上でのVLIW命令の配置例を示す図



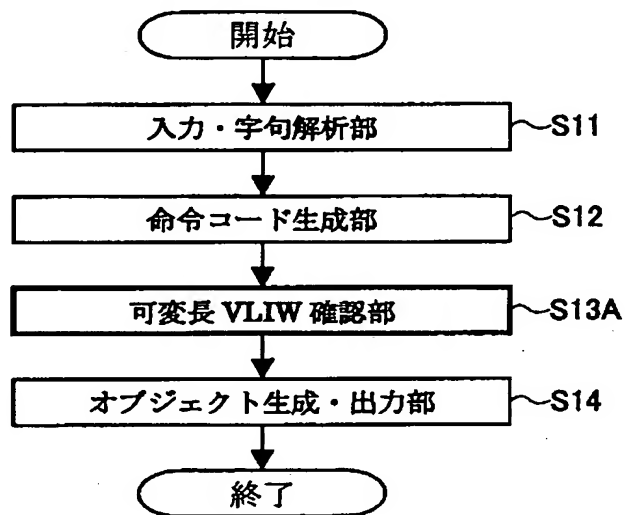
【図 13】

図10に示す命令レジスタにVLIW命令が格納される様子の一例を示す図



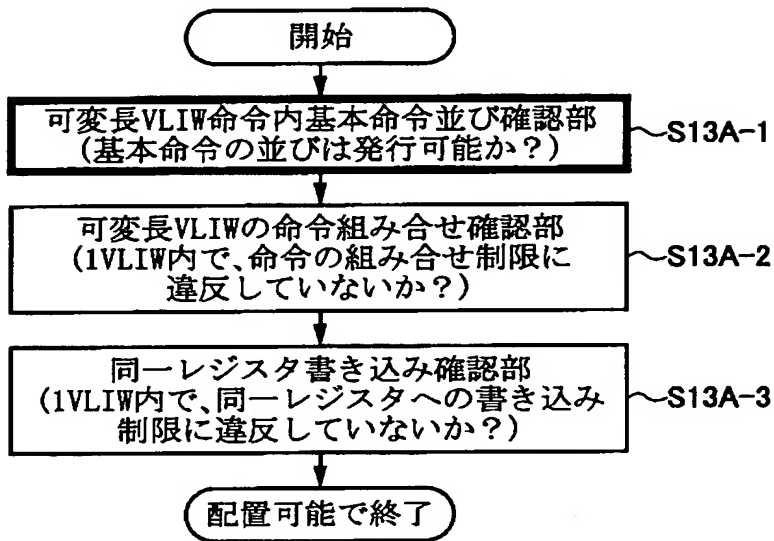
【図 14】

本発明の一実施の形態による可変長VLIWプロセッサの
アセンブラのフローチャート



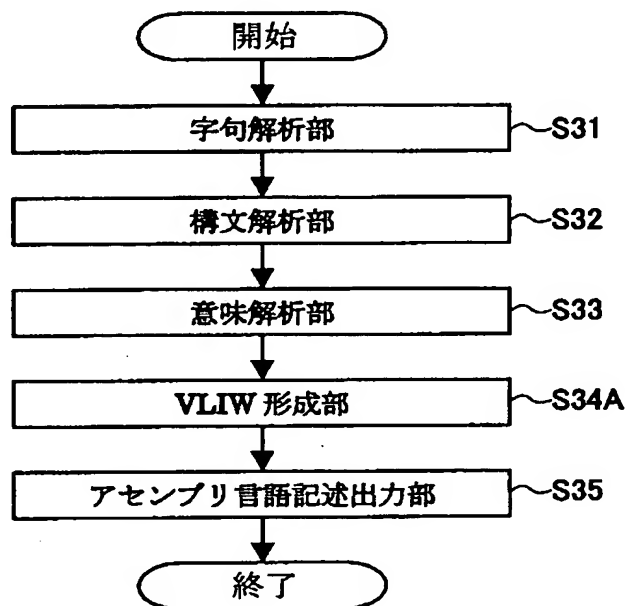
【図 1 5】

図13に示す可変長VLIW確認部の
一構成例を示すフローチャート



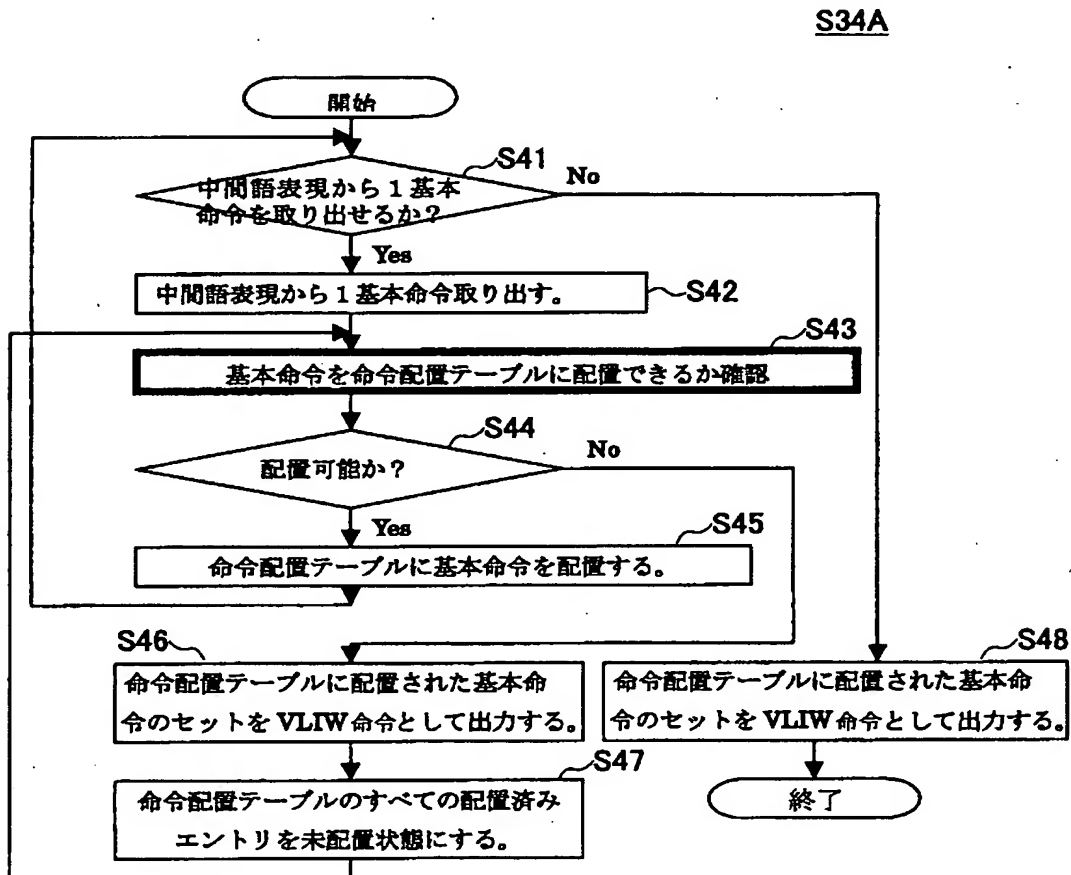
【図 1 6】

本発明の一実施の形態による可変長VLIWプロセッサの
コンパイラのフローチャート



【図 17】

図6に示す可変長形成部の一構成例を示すフローチャート



【図 18】

命令格納テーブルの一例を示す図

基本命令	機能ユニット種別
ADD	IU
FADD	FU
BRA	BU

IU:整数ユニットにて実行可能

FU:浮動小数点数ユニットにて実行可能

BU:分岐ユニットにて実行可能

【図 19】

命令スロットテーブルの一例を示す図

論理命令 スロット	配置可能な 機能ユニット種別		
	IU	FU	BU
0	○	—	—
1	—	○	—
2	○	—	—
3	—	○	—
4	—	—	○
5	—	—	○

○:実行可、—:実行不可

【図 2 0】

命令配置テーブルの一例を示す図

物理命令スロット	配置された 命令の情報
命令スロット 0	—
命令スロット 1	—
命令スロット 2	—
命令スロット 3	—

— : 未配置、○ : 配置済

【図 2 1】

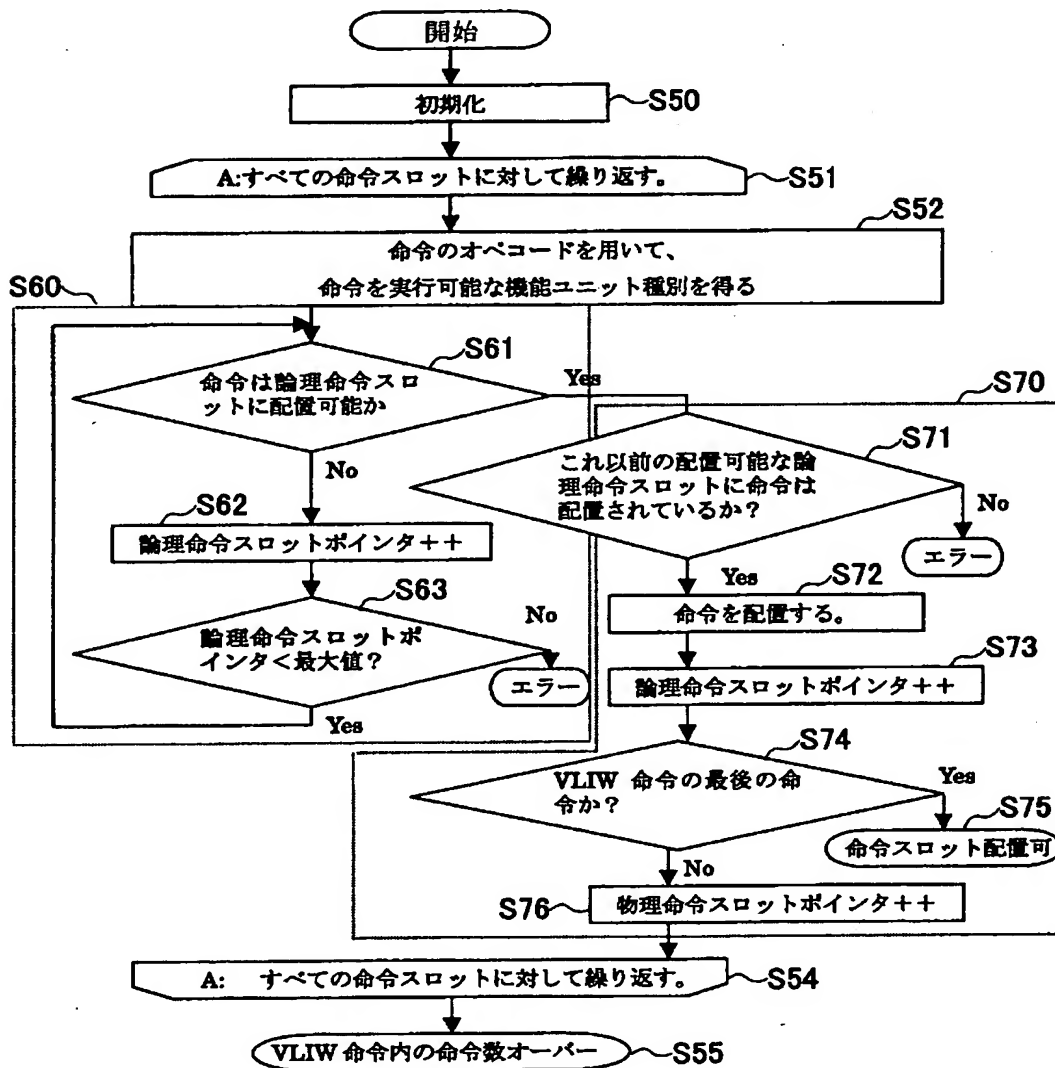
機能ユニットテーブルの一例を示す図

機能ユニット	命令が配置された機能ユニットの機能ユニット番号のビットパターン			
	0	1	2	3
IU	—	—	—	—
FU	—	—	—	—
BU	—	—	—	—

○ : 配置済、 — : 未配置

【図 22】

可変長VLIW命令を構成する基本命令の並びを確認する
第1のアルゴリズムを示すフローチャート



【図 23】

図16に示す可変長形成部の第1の構成例における
命令種別テーブルを示す図

命令オペ コード	機能ユニット種別	機能ユニット種別の 機能ユニット番号			
		0	1	2	3
ADD	IU	○	○	—	—
FADD	FU	○	○	—	—
BRA	BU	○	○	—	—

IU:整数ユニットにて実行可能

FU:浮動小数点数ユニットにて実行可能

BU:分岐ユニットにて実行可能

○:配置可

—:配置不可

【図 24】

図16に示す可変長形成部の第1の構成例における
命令スロットテーブルを示す図

論理命令 スロット	配置可能な 機能ユニット種別			機能ユニット種別毎 の機能ユニット番号			
	IU	FU	BU	0	1	2	3
0	○	—	—	○	—	—	—
1	—	○	—	○	—	—	—
2	○	—	—	—	○	—	—
3	—	○	—	—	○	—	—
4	—	—	○	○	—	—	—
5	—	—	○	—	○	—	—

○:配置可

—:配置不可

【図 2 5】

可変長VLIW命令を構成する基本命令の
並びを確認する第2のアルゴリズムで用いられる
命令種別テーブルの一例を示す図

基本命令	命令カテゴリ
ADD	I_1
DIV	I_2
FADD	F_1
BRA	B_1

I_1 : 整数カテゴリ 1

I_2 : 整数カテゴリ 2

F_1 : 浮動小数点数カテゴリ 1

B_1 : 分岐カテゴリ 1

【図 2 6】

可変長VLIW命令を構成する基本命令の
並びを確認する第2のアルゴリズムで用いられる
命令カテゴリテーブルの一例を示す図

命令カテ ゴリ	機能ユニット 種別	機能ユニット種別毎 の機能ユニット番号 のビットパターン			
		0	1	2	3
I_1	IU	○	○	—	—
I_2	IU	○	—	—	—
F_1	FU	○	○	—	—
B_1	BU	○	—	—	—

○ : 実行可、— : 実行不可

【図 2 7】

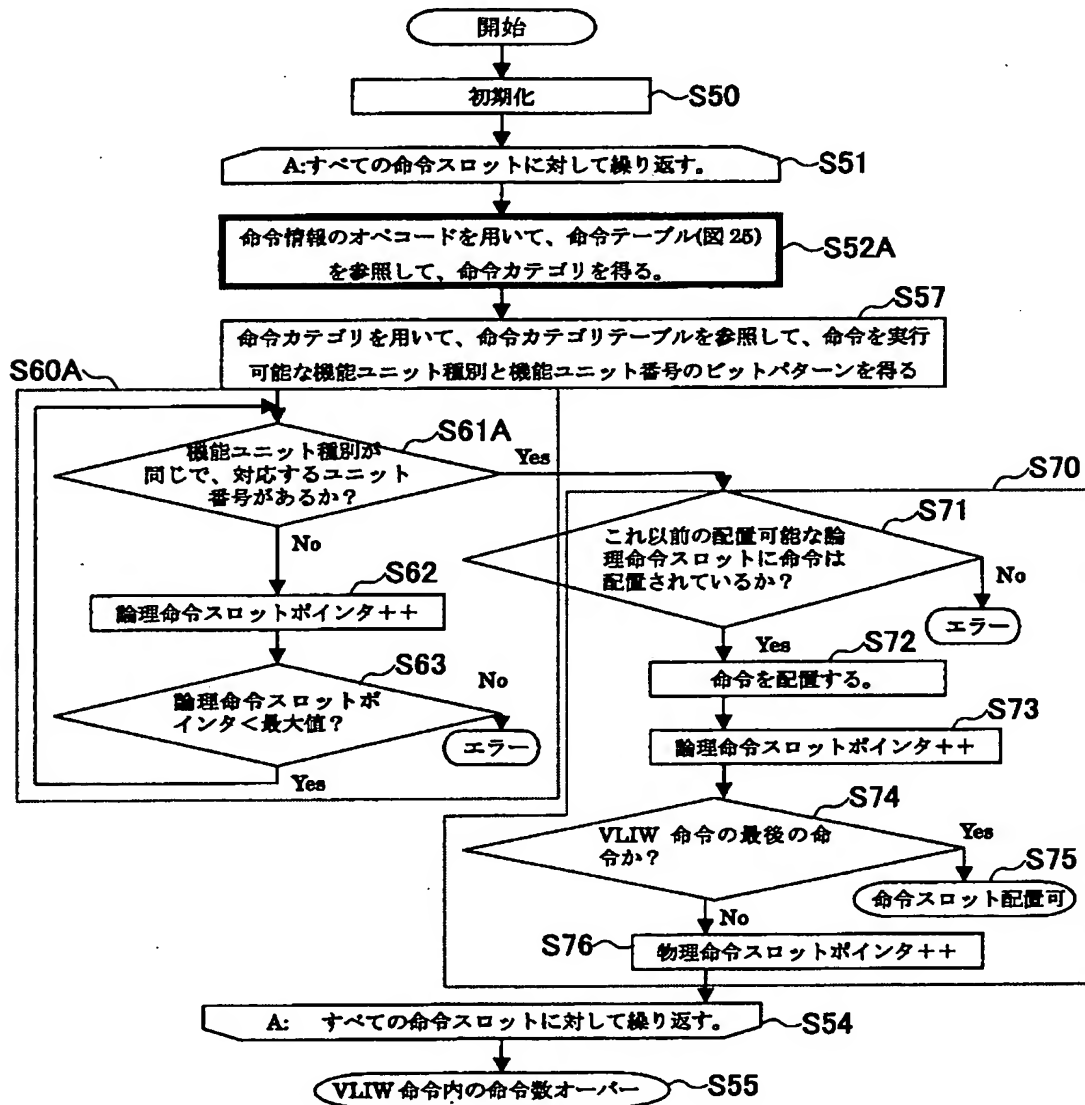
可変長VLIW命令を構成する基本命令の
並びを確認する第2のアルゴリズムで用いられる
命令スロットテーブルの一例を示す図

論理命令 スロット	配置可能な 機能ユニット種別			機能ユニット種別毎 の機能ユニット番号の ビットパターン			
	IU	FU	BU	0	1	2	3
0	○	—	—	○	—	—	—
1	—	○	—	○	—	—	—
2	○	—	—	—	○	—	—
3	—	○	—	—	○	—	—
4	—	—	○	○	—	—	—
5	—	—	○	—	○	—	—

○：実行可、—：実行不可

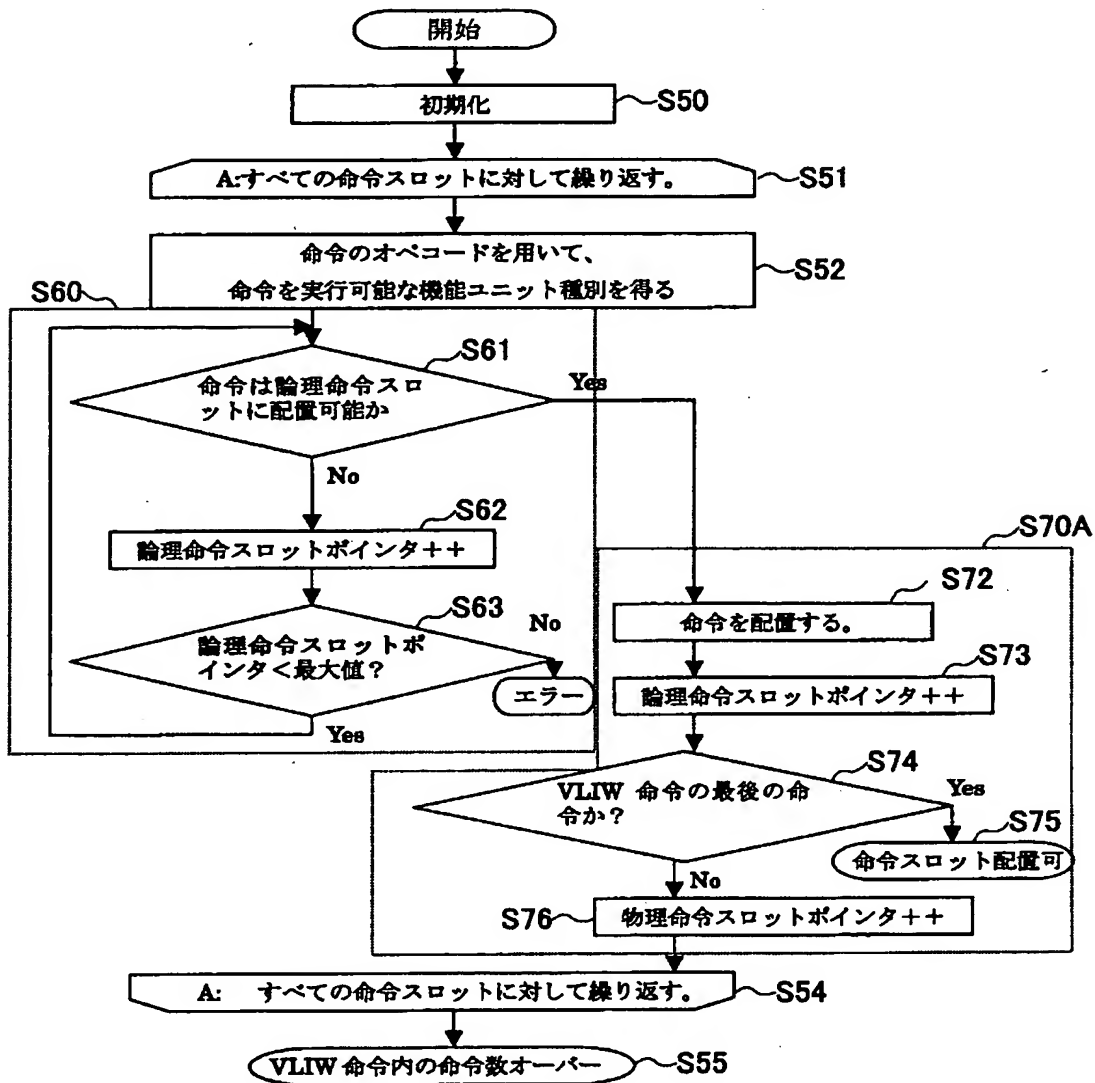
【図 28】

可変長VLIW命令を構成する基本命令の並びを確認する
第2のアルゴリズムを示すフローチャート



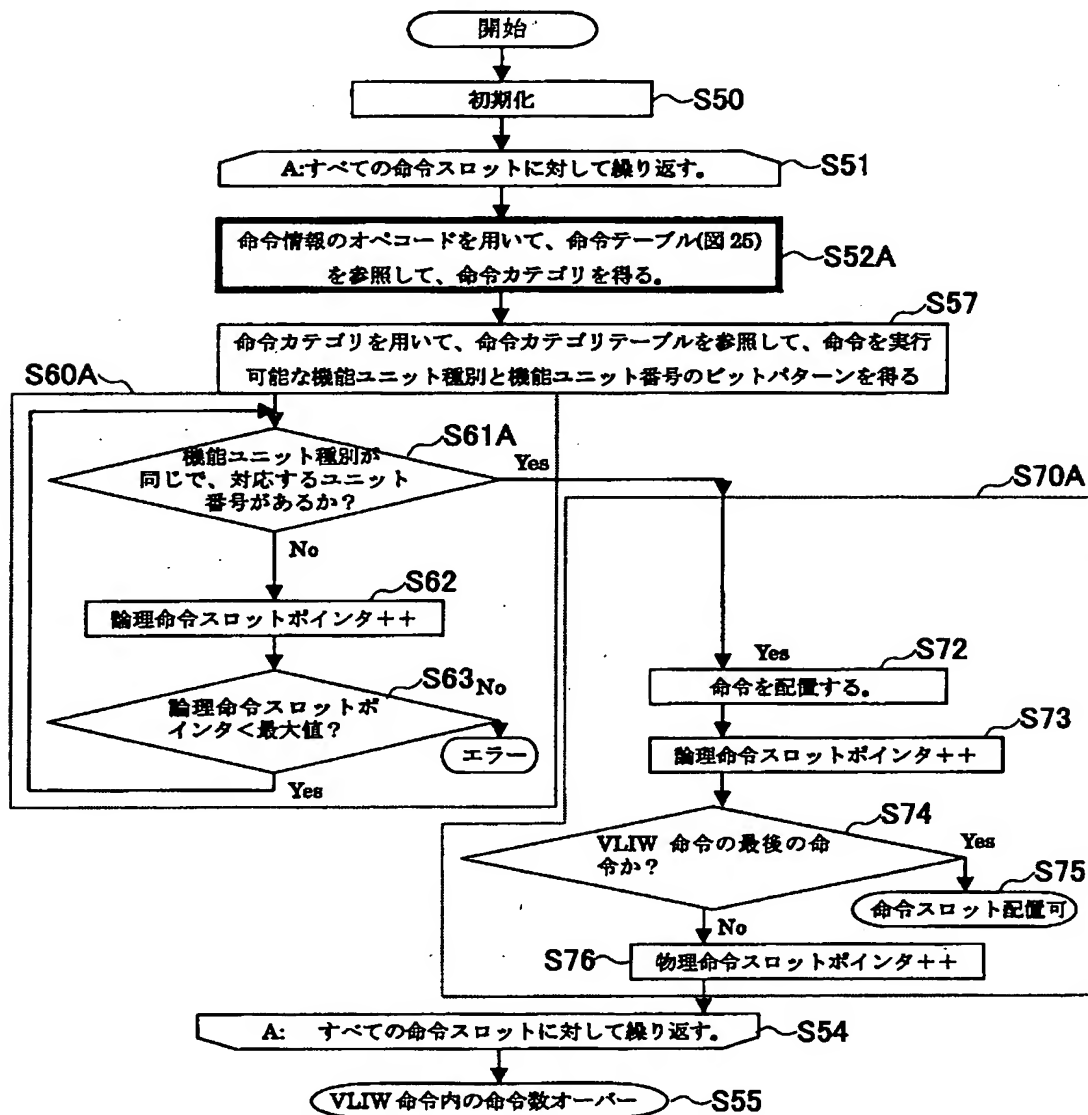
【図 29】

可変長VLIW命令を構成する基本命令の並びを確認する
第3のアルゴリズムを示すフローチャート



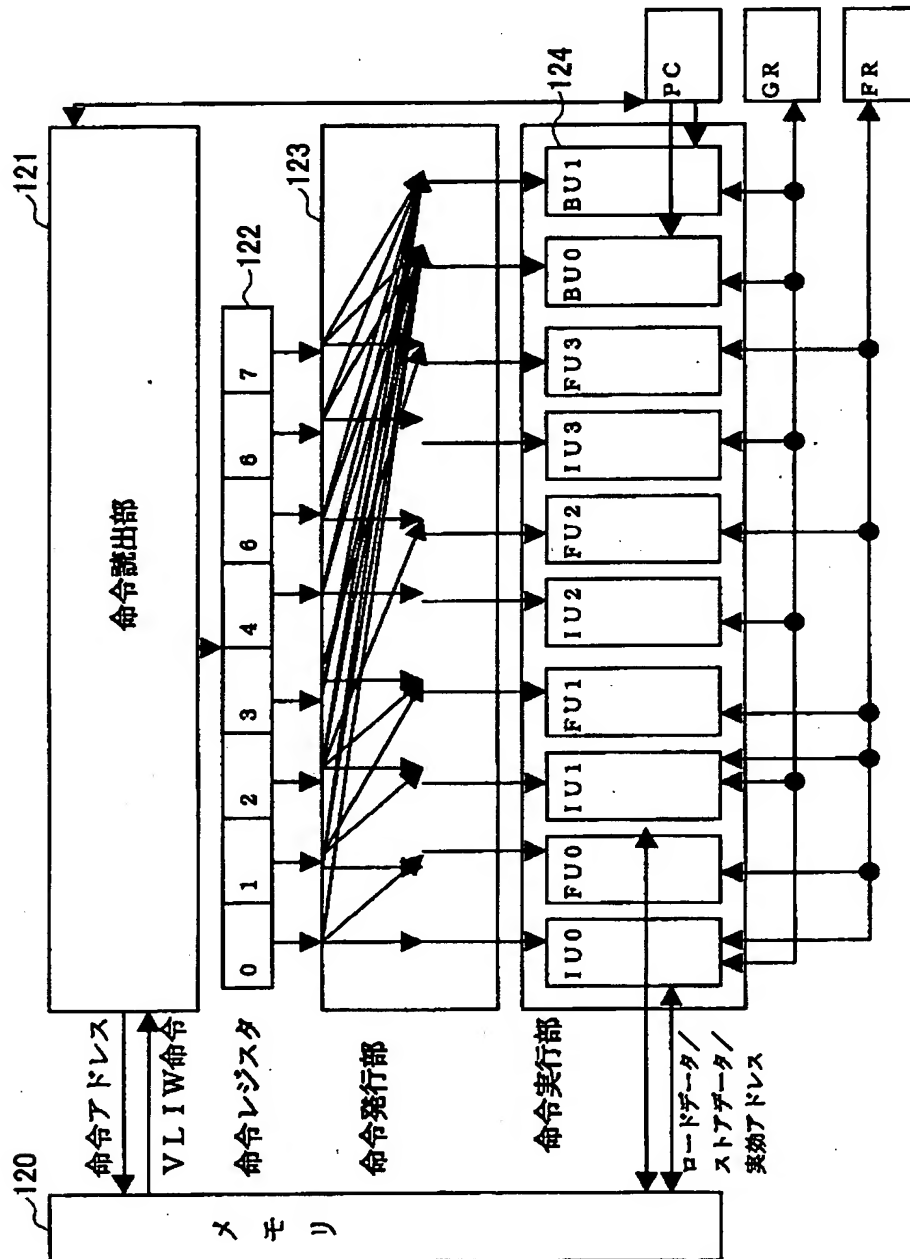
【図 30】

可変長VLIW命令を構成する基本命令の並びを確認する
第4のアルゴリズムを示すフローチャート



【図 31】

可変長VLWプロセッサの第2の構成例を示すブロック図



【図 3 2】

図31に示す可変長VLIWプロセッサで実行可能な
VLIW内の基本命令の並びを示す図(その1)

命令スロ ット0	命令スロ ット1	命令スロ ット2	命令スロ ット3	命令スロ ット4	命令スロ ット5	命令スロ ット6	命令スロ ット7
I0	F0	I1	F1	I2	F2	I3	F3
I0	F0	I1	F1	I2	F2	I3	B0
I0	F0	I1	F1	I2	F2	I3	
I0	F0	I1	F1	I2	F2	F3	B0
I0	F0	I1	F1	I2	F2	F3	
I0	F0	I1	F1	I2	F2	B0	B1
I0	F0	I1	F1	I2	F2	B0	
I0	F0	I1	F1	I2	F2		
I0	F0	I1	F1	I2	I3	B0	B1
I0	F0	I1	F1	I2	I3	B0	
I0	F0	I1	F1	I2	I3		
I0	F0	I1	F1	F2	F3	B0	B1
I0	F0	I1	F1	F2	F3	B0	
I0	F0	I1	F1	F2	F3		
I0	F0	I1	F1	I2	B0	B1	
I0	F0	I1	F1	I2	B0		
I0	F0	I1	F1	I2			
I0	F0	I1	F1	F2	B0	B1	
I0	F0	I1	F1	F2	B0		
I0	F0	I1	F1	F2			
I0	F0	I1	I2	I3	B0	B1	
I0	F0	I1	I2	I3	B0		
I0	F0	I1	I2	I3			
I0	F0	F1	F2	F3	B0	B1	
I0	F0	F1	F2	F3	B0		
I0	F0	F1	F2	F3			
I0	F0	I1	F1	B0	B1		
I0	F0	I1	F1	B0			
I0	F0	I1	F1				
I0	F0	I1	I2	B0	B1		
I0	F0	I1	I2	B0			
I0	F0	I1	I2				
I0	F0	F1	F2	B0	B1		
I0	F0	F1	F2	B0			
I0	F0	F1	F2				
I0	F0	I1	I2	B0	B1		
I0	F0	I1	I2	B0			
I0	F0	I1	I2				
F0	F1	F2	F3	B0	B1		
F0	F1	F2	F3	B0			

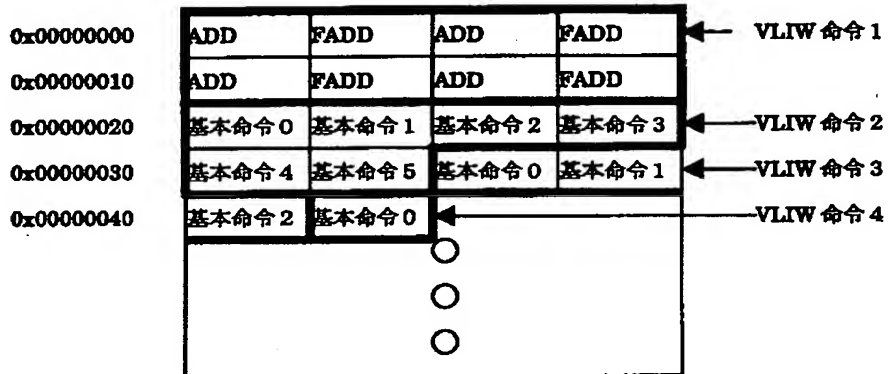
【図 33】

図31に示す可変長VLIWプロセッサで実行可能な
VLIW内の基本命令の並びを示す図(その2)

F0	F1	F2	F3				
I0	I1	I2	I3	B0	B1		
I0	I1	I2	I3	B0			
I0	I1	I2	I3				
I0	F0	I1	B0	B1			
I0	F0	I1	B0				
I0	F0	I1					
I0	F0	F1	B0	B1			
I0	F0	F1	B0				
I0	F0	F1					
F0	F1	F2	B0	B1			
F0	F1	F2	B0				
F0	F1	F2					
I0	I1	I2	B0	B1			
I0	I1	I2	B0				
I0	I1	I2					
I0	F0	B0	B1				
I0	F0	B0					
I0	F0						
F0	F1	B0	B1				
F0	F1	B0					
F0	F1						
I0	I1	B0	B1				
I0	I1	B0					
I0	I1						
I0	B0	B1					
I0	B0						
I0							
F0	B0	B1					
F0	B0						
F0							
B0	B1						
B0							

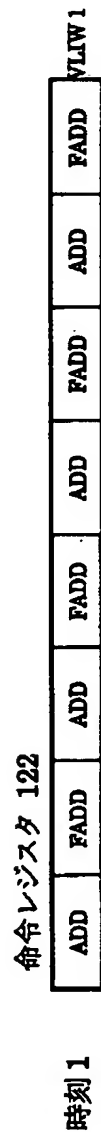
【図 34】

図31に示すメモリ上でのVLIW命令の配置例を示す図



【図 3 5】

図31に示す命令レジスタにVLIW命令が格納される様子を示す図



【図 3 6】

図31に示す第2の構成例における
命令種別テーブルを示す図

命令オペ コード	機能ユニット種別	機能ユニット種別の 機能ユニット番号			
		0	1	2	3
ADD	IU	○	○	○	○
FADD	FU	○	○	○	○
BRA	BU	○	○	—	—

IU:整数ユニットにて実行可能

FU:浮動小数点数ユニットにて実行可能

BU:分岐ユニットにて実行可能

○:配置可

—:配置不可

【図 3 7】

図31に示す第2の構成例における
命令スロットテーブルを示す図

論理命令 スロット	配置可能な 機能ユニット種別			機能ユニット種別毎 の機能ユニット番号			
	IU	FU	BU	0	1	2	3
0	○	—	—	○	—	—	—
1	—	○	—	○	—	—	—
2	○	—	—	—	○	—	—
3	—	○	—	—	○	—	—
4	○	—	—	—	—	○	—
5	—	○	—	—	—	○	—
6	○	—	—	—	—	—	○
7	—	○	—	—	—	—	○
8	—	—	○	○	—	—	—
9	—	—	○	—	○	—	—

○：配置可

—：配置不可

【図 3 8】

可変長VLIWプロセッサの第3の構成例における
命令種別テーブルを示す図

命令オペ コード	命令カテゴリ
ADD	I_1
FADD	F_1
BRA	B_1

I_1：整数カテゴリ 1

F_1：浮動小数点数カテゴリ 1

B_1：分岐カテゴリ 1

【図 3 9】

可変長VLWプロセッサの第3の構成例における 命令カテゴリーテーブルを示す図

命令カテ ゴリ	機 能 ユ ニ ッ ト 種 別	機能ユニット種別の 機能ユニット番号			
		0	1	2	3
I_1	IU	○	○	—	—
F_1	FU	○	○	—	—
B_1	BU	○	○	—	—

○：実行可、—：実行不可

【図 4 0】

可変長VLWプロセッサの第3の構成例における 命令スロットテーブルを示す図

論理命令 スロット	配置可能な 機能ユニット種別			機能ユニット種別毎 の機能ユニット番号			
	IU	FU	BU	0	1	2	3
0	○	—	—	○	—	—	—
1	—	○	—	○	—	—	—
2	○	—	—	—	○	—	—
3	—	○	—	—	○	—	—
4	—	—	○	○	—	—	—
5	—	—	○	—	○	—	—

○：実行可、—：実行不可

【図 4 1】

可変長VLIWプロセッサの第4の構成例における 命令種別テーブルを示す図

命令オペ コード	命令カテゴリ
ADD	I_1
FADD	F_1
BRA	B_1

I_1: 整数カテゴリ 1

F_1: 浮動小数点数カテゴリ 1

B_1: 分岐カテゴリ 1

【図 4 2】

可変長VLIWプロセッサの第3の構成例における 命令カテゴリテーブルを示す図

命令カテ ゴリ	機能ユニット 種別	機能ユニット種別の 機能ユニット番号			
		0	1	2	3
I_1	IU	○	○	—	—
F_1	FU	○	○	—	—
B_1	BU	○	○	—	—

○: 配置可

—: 配置不可

【図 4 3】

可変長VLWプロセッサの第4の構成例における
命令スロットテーブルを示す図

論理命令 スロット	配置可能な 機能ユニット種別			機能ユニット種別毎 の機能ユニット番号			
	IU	FU	BU	0	1	2	3
0	○	—	—	○	—	—	—
1	—	○	—	○	—	—	—
2	○	—	—	—	○	—	—
3	—	○	—	—	○	—	—
4	○	—	—	—	—	○	—
5	—	○	—	—	—	○	—
6	○	—	—	—	—	—	○
7	—	○	—	—	—	—	○
8	—	—	○	○	—	—	—
9	—	—	○	—	○	—	—

○：配置可

—：配置不可

【書類名】 要約書

【要約】

【課題】 可変長の超長命令語アーキテクチャに基づいて命令発行幅が異なるプロセッサの言語処理系に適した命令処理方法を提供する。

【解決手段】 プロセッサが実行する命令を構成する基本命令を実行できる機能ユニットの種別を特定する第1のステップと、基本命令を論理命令スロットに配置可能かどうかを判断する第2のステップと、配置可能と判断された基本命令と論理命令スロットに配置される他の基本命令との関係を考慮して当該配置可能と判断された基本命令をプロセッサの命令スロットに配置する第3のステップとを設けた。

【選択図】 図 2 2

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社